

Contents

1	Overview.....	1
1.1	Kernel Diagnostics.....	1
1.2	Self-Test Diagnostics.....	1
1.3	Extended Diagnostics.....	1
1.4	Running Self-Tests/Extended Diagnostics From Normal Operation.....	2
1.5	Front Panel.....	2
1.6	Clearing NVRAM.....	2
1.7	Diagnostic Options Jumpers.....	3
1.7.1	Executive.....	3
1.7.2	Acquisition.....	4
1.8	Equipment Required.....	5
1.9	Circuit Coverage.....	6
2	Kernel Diagnostics.....	7
2.1	Error Index Overview.....	7
2.2	Test Status Indicators.....	8
2.3	Start/Stop Trigger Pin or LED.....	8
2.4	Executive Kernel.....	8
2.4.1	Power-Up Options.....	10
2.5	Display Kernel.....	11
2.6	Timebase Kernel.....	13
2.6.1	Miscellaneous Non-Kernel LED Patterns.....	15
2.6.1.1	Diagnostic Operation.....	15
2.6.1.2	Normal Operation.....	16
2.7	Acquisition Kernel.....	16
2.7.1	Power-Up Options.....	18
2.7.2	Miscellaneous Non-Kernel LED Patterns.....	18
2.7.2.1	Diagnostic Operation.....	18
2.7.2.2	Normal Operation.....	19
2.8	Power-Up Subsystem Communications.....	19
2.8.1	Executive Subsystem Routines.....	19
2.8.2	Indications of Subsystem Communication Failures.....	19
2.8.3	Troubleshooting A Failed Subsystem Communication Path.....	20
2.8.3.1	Executive to Display or Timebase.....	20
2.8.3.2	Timebase to Acquisition.....	20
3	Troubleshooting Non-Functioning Subsystem Kernels.....	22
3.1	Equipment Required.....	22
3.2	Executive Kernel.....	22
3.2.1	Kernel Extender Service Kit.....	23
3.2.2	Using the Kernel Extender Card.....	23
3.2.2.1	Mode 1 -- Passive Extender.....	24
3.2.2.2	Mode 2 -- Active Kernel Exerciser.....	24
3.2.2.3	Overhead Cable Replacement.....	25

Contents

3.2.2.4	Extender Card Test Points	26
3.2.2.5	Extender Card Connectors.....	26
3.2.3	Using the RS-232-C Loopback Device.....	29
3.3	Display Kernel.....	29
3.3.1	Setting Kernel Test Jumpers.....	29
3.3.2	Kernel Test Address Sequence.....	30
3.3.3	Logic Analyzer Set-up.....	31
3.4	Timebase Kernel.....	31
3.4.1	Setting the Kernel Test Jumper	31
3.4.2	Kernel Test Address Sequence.....	32
3.4.3	Logic Analyzer Set-up.....	32
3.5	Acquisition Kernel.....	32
3.5.1	Setting Kernel Test Jumpers.....	33
3.5.2	Kernel Test Address Sequence.....	33
3.5.3	Logic Analyzer Set-up.....	34
4	Self-Test Diagnostics.....	35
5	Extended Diagnostics.....	36
5.1	Error Index Overview.....	36
5.2	Extended Diagnostics Menu Displays.....	37
5.2.1	Subsys(tem) Menu.....	39
5.2.1.1	Prompt/Warning Section.....	39
5.2.1.2	Test Status Section.....	39
5.2.1.3	Control Section.....	40
5.2.1.4	Front Panel Buttons & Knobs	42
5.2.2	Block Menu.....	42
5.2.2.1	Test Status Section.....	44
5.2.3	Area Menu.....	44
5.2.3.1	Test Status Section.....	46
5.2.4	Routine Menu.....	46
5.2.4.1	Test Status Section.....	48
5.2.4.2	Control Section.....	48
5.2.4.2.1	Interaction With Test Status Selectors.....	48
5.2.4.2.2	Special Combinations - Cycle & Cycle-Halt Modes.....	49
5.2.5	Help Menu.....	49
5.2.5.1	Front Panel Buttons	50
5.2.6	Prompt/Keypad Menus.....	50
5.2.6.1	Front Panel Buttons	52
5.2.7	Debugger Menu.....	52
5.2.7.1	Action Editor.....	55
5.2.7.2	List Editor.....	56
5.2.7.3	Front Panel Buttons	58
5.2.8	Other Displays.....	59

Contents

5.3	Audible Indication of Test Termination	59
5.4	Front Panel Buttons, Knobs, and LEDs	59
5.5	Hardcopy Operation	60
5.6	Remote Diagnostics	61
5.6.1	Front Panel Emulation Mode (RS-232-C Only)	61
5.6.1.1	Entry	61
5.6.1.2	Exit	62
5.6.1.3	Non-Displayed Commands	62
5.6.2	System Mode (GPIB and RS-232-C)	63
5.6.2.1	TEST & TEST XTND Commands	63
5.6.2.2	TEST MAN Command	64
5.6.2.3	DIAG? Query Command	64
5.6.2.3.1	Diagnostics Passed Response	64
5.6.2.3.2	Diagnostics Failed Response	65
5.6.2.3.3	Diagnostics Bypassed Response	65
6	Test Descriptions	66
6.1	Kernel Page Layout	66
6.2	Self-Test/Extended Page Layout	66
6.3	Notation Used	67
6.4	Executive Kernel Overview	68
6.5	Display Kernel Overview	69
6.6	Timebase Kernel Overview	69
6.7	Acquisition Kernel Overview	69
6.8	Executive Self-Test/ Extended Overview	70
6.9	Display Self-Test/ Extended Overview	74
6.10	Timebase Self-Test/ Extended Overview	75
6.11	Mainframe & SM-11 Acquisition Self-Test/ Extended Overview	78

List of Tables

TABLE 1-1 Jumper Settings of the I/O Board Diagnostic Options Jumpers.....	4
TABLE 1-2 Jumper Settings of the Acquisition Diagnostic Options Jumpers	5
TABLE 2-1 Executive Kernel Test Mode LEDs.....	10
TABLE 2-2 Executive Kernel Status Pins & MENUS LEDs Encoding	10
TABLE 2-3 Display Kernel Test Mode LEDs.....	13
TABLE 2-4 Timebase Kernel Test Mode LEDs.....	15
TABLE 2-5 Timebase (Diagnostics Operation) Non-Kernel LED Patterns.....	16
TABLE 2-6 Timebase (Normal Operation) Non-Kernel LED Patterns.....	16
TABLE 2-7 Acquisition Kernel Test Mode LEDs.....	17
TABLE 2-8 Acquisition (Diagnostics Operation) Non-Kernel LED Patterns.....	18
TABLE 2-9 Timebase (Normal Operation) Non-Kernel LED Patterns.....	19
TABLE 3-1 Overhead Cable Replacement Guide.....	26
TABLE 3-2 Extender Board Test Signal Connector Pinout.....	27
TABLE 5-1 Cycle and Cycle-Halt Mode Set-ups.....	49
TABLE 5-2 Summary of Menus vs. Active Buttons.....	60

List of Figures

Figure 1-1. Location of the Diagnostic Options Jumpers on the A14 Input/Output Board.....	3
Figure 1-2. Location of the Diagnostic Options Jumpers on the A25/A28 Acquisition MPU Board	5
Figure 2-1. A17 Main Processor Board Status LEDs & Pins.....	9
Figure 2-2. A7 Display Controller Board Status LEDs & Pins.....	12
Figure 2-3. A5 Timebase Controller Board Status LEDs	14
Figure 2-4. A25/A28 Acquisition MPU Board Status LEDs.....	17
Figure 5-1. Subsystem Menu Example	38
Figure 5-2. Block Menu Example.....	43
Figure 5-3. Area Menu Example.....	45
Figure 5-4. Routine Menu Example.....	47
Figure 5-5. Prompt/Keypad Screen Layout.....	51
Figure 5-6. Debugger Menu Example.....	54

1 Overview

Each time the front panel ON/STANDBY switch is set to ON, the 11801 (or 11802) and attached SM-11 Multi-Channel Unit's perform Kernel Diagnostics on their respective processor subsystems, followed by Self-Test Diagnostics on most all of their major circuits. These tests run automatically and return the instrument to normal operation when no faults are found.

NOTE

Pressing front panel buttons, turning knobs, or touching the Touch Panel while tests are running may cause a diagnostic failure.

If any Self-Test failures occur, the instrument emits a double high-low beep. It then enters the Extended Diagnostics mode and displays a menu showing the status of all major subsystems. Subsystems with Self-Test failures will have an error index code next to the subsystem name. For Kernel Diagnostic test failures, a different type of error index codes are read using procedures described later in this section.

Before the power-up Self-Tests can run, the power supply must be operating properly. Power supply troubleshooting is discussed in documentation provided with the Extended Diagnostics 11000-Series Power Supplies Troubleshooting Fixture (067-1264-00).

1.1 Kernel Diagnostics

At power-up, and only at power-up, the Executive, Display, Timebase, Mainframe Acquisitions, and all resident SM-11 Multi-Channel Unit Acquisition processors concurrently run a local set of Kernel Diagnostics tests. These tests verify hardware critical to diagnostic operation, such as ROM, RAM, DMA's, timers, and interrupt control circuitry. Processors that can not run Kernel Diagnostics (i.e. can not execute instructions from ROM) can be troubleshot with the procedures described under Troubleshooting Non-Functioning Subsystem Kernels. Failure of a Kernel Diagnostic test causes the failing subsystem to loop on the failed test, indicating its failure on internal circuit board LEDs and status pins. (The Executive and Display have additional error reporting facilities, namely the front panel LEDs and the display screen.)

1.2 Self-Test Diagnostics

If the Executive subsystem runs all of its Kernel Diagnostics without failure it will start Self-Test Diagnostics (possibly only locally), even though the Display and/or Timebase may have failed to establish communication with the Executive subsystem. The Self-Test Diagnostics, which are run at power-up and can also be run at other times, help to verify overall instrument functionality in a short period of time.

1.3 Extended Diagnostics

Any failures in the Self-Test Diagnostics cause the instrument to enter the Extended Diagnostics mode, after the remaining Self-Tests have run. The Extended Diagnostics display shows a test menu with the names of major subsystems and the test status of each. The test status will be either a "pass" indicator, a "****" indicator for tests that have not run or which do not have current status information, a "???"

indicator for optional hardware which is not present or subsystems whose communication paths are not functional, or a five-digit error index code that refers to a test description in this manual. See the Error Index Overview section under Extended Diagnostics on how to access a test description for a given error index code.

Extended Diagnostics consists of all Self-Tests plus a number of interactive tests used to troubleshoot, verify, or calibrate specific circuits. In addition, there may be some automatic tests which were excluded from being run in Self-Test, such as NVRAM tests or lengthy RAM tests, to protect the instrument from undesirable events on power-up or to reduce the Self-Test execution time. In the Extended Diagnostics mode, one or more tests can be selected and run with user-selectable test conditions. These tests can also be run remotely with a terminal/controller connected to the RS-232-C or GPIB interface. See the Remote Diagnostics section under Extended Diagnostics for detailed information on external diagnostics control.

1.4 Running Self-Tests/Extended Diagnostics From Normal Operation

Self-Test or Extended Diagnostics can be selected and run at any time during normal instrument operation. To run the tests, press the UTILITY button, select either the Self-Test or the Extended Diagnostic label, and then confirm the selection by selecting the YES label from the subsequent pop-up. When the Self-Tests are invoked in this way, the instrument will not run the Kernel Diagnostics. Self-Test Diagnostics contain equivalent Kernel Diagnostic tests, therefore circuit coverage is not being reduced by not performing the actual Kernel Diagnostic tests. All Self-Tests will run to verify instrument functionality, then return the instrument to normal operation if no faults occur. Once again, a failure of the Self-Tests will automatically invoke the Extended Diagnostics mode.

1.5 Front Panel

While in Self-Test, normal functions of the front panel are suspended. Pressing a front panel button, turning a knob, or touching the Touch Panel may cause an apparent failure. During Extended Diagnostics, several of the front panel buttons become functional, in addition to the Touch Panel and knobs. For instance, when an Extended Diagnostics menu is displayed, you can press the front panel Hardcopy button for a permanent record of test failures (a compatible printer must be connected). See the Extended Diagnostics section for further information on when buttons and knobs are active and their function.

1.6 Clearing NVRAM

Before a power-up Self-Test begins -- but just after the Executive processor has run its Kernel Diagnostics -- the front panel buttons are scanned by the Executive processor. If the Executive processor senses that the Waveform and Trigger buttons, and only these two buttons, are pressed in (i.e. closed) during this time, then the Executive processor resets its NVRAM to a default state. This essentially destroys all stored settings and saved trace descriptions (there are no stored waveforms in NVRAM). When this occurs, the NVRAM is initialized by filling all but a few locations with a default value. The following items are left intact after the NVRAM is reset:

- Number of instrument power-ons (POWERON?)
- Instrument power-on time (UPTIME?)
- Mainframe serial number (UID? MAIN)

1.7 Diagnostic Options Jumpers

1.7.1 Executive

Executive Diagnostic Options jumpers J710 - J715 on the A14 Input/Output Board (see Figure 1-1 below) provide a way to separately enable or disable the Kernel Diagnostics (Executive Subsystem only), Self-Tests, and Extended Diagnostics at power-up. Also, the default RS-232-C baud rate for external control of the Diagnostics can be changed from the factory setting of 9600. One jumper setting for the Kernel Diagnostics causes the instrument to continuously loop on the RS-232 loopback test to aid in troubleshooting the RS-232-C port. For some functions, two jumpers are used together to specify the setting. Note that only jumper J710 is actually labeled on the circuit board.

Each jumper is set (true) when the two pins on the right are shorted together. In Table 1-1, a set jumper is indicated by a "1" and a non-set (false) jumper by a "0." Factory settings are indicated by a "Normal" entry in the Description column.

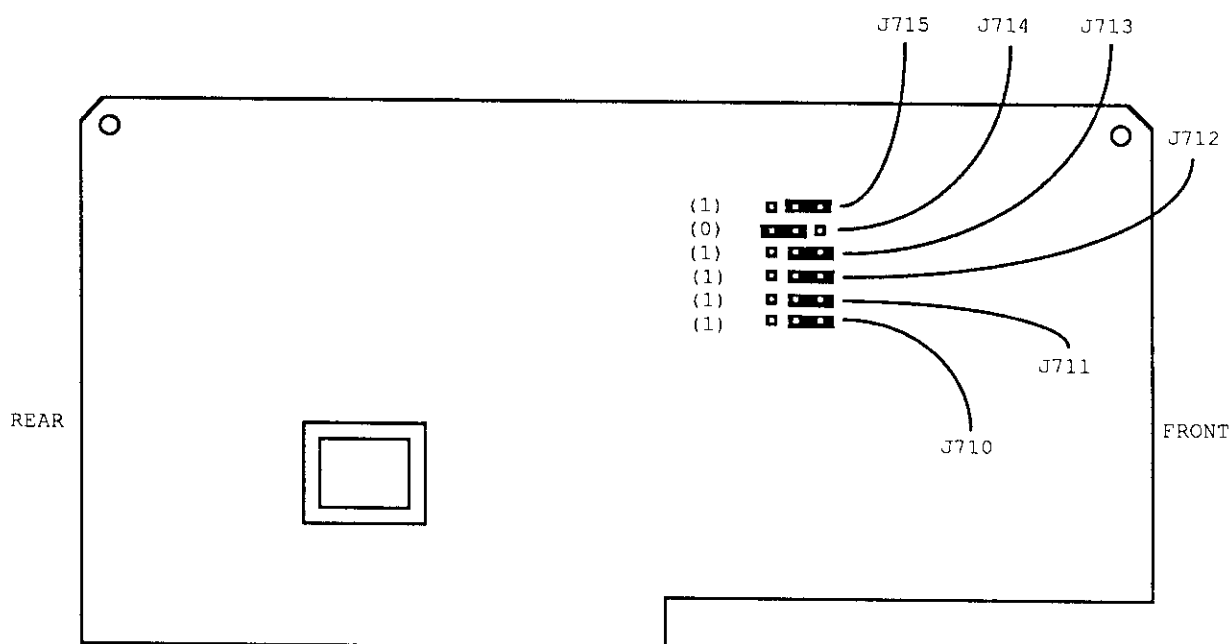


Figure 1-1. Location of the Diagnostic Options Jumpers on the A14 Input/Output Board

TABLE 1-1
Jumper Settings of the I/O Board Diagnostic Options Jumpers

<u>Controlled Parameter</u>	<u>Jumper(s)</u>	<u>Description</u>
Executive Kernel Diagnostics:	J711 J710	
	0 0	Bypass Kernel Diagnostics.
	0 1	Loop on RS-232 test after running Kernel Diagnostics.
	1 0	Run Kernel Diagnostics but do not loop on a failed test.
	1 1	<i>Normal;</i> run Kernel Diagnostics and loop on the first failed test.
Self-Test Diagnostics:	J712	
	0	Bypass Self-Test Diagnostics at power-up.
	1	<i>Normal;</i> run Self-Tests at power-up.
Extended Diagnostics:	J713	
	0	Bypass Extended Diagnostics after any Self-Test.
	1	<i>Normal;</i> run Extended Diagnostics after a Self-Test failure.
RS-232-C Baud Rate during Extended Diagnostics:	J715 J714	
	0 0	300 baud.
	0 1	1200 baud.
	1 0	<i>Normal;</i> 9600 baud.
	1 1	19,200 baud.

1.7.2 Acquisition

Acquisition Diagnostic Options jumpers J111, J210, and J110 on the A25/A28 Acquisition MPU Board (see Figure 1-2 below) provide a way to bypass Acquisition calibration or invoke an interface arbiter test loop on power-up. Refer to Troubleshooting A Failed Subsystem Communication Path for further

information concerning the arbitor test loop. Note that none of these jumpers are actually labelled on the circuit board.

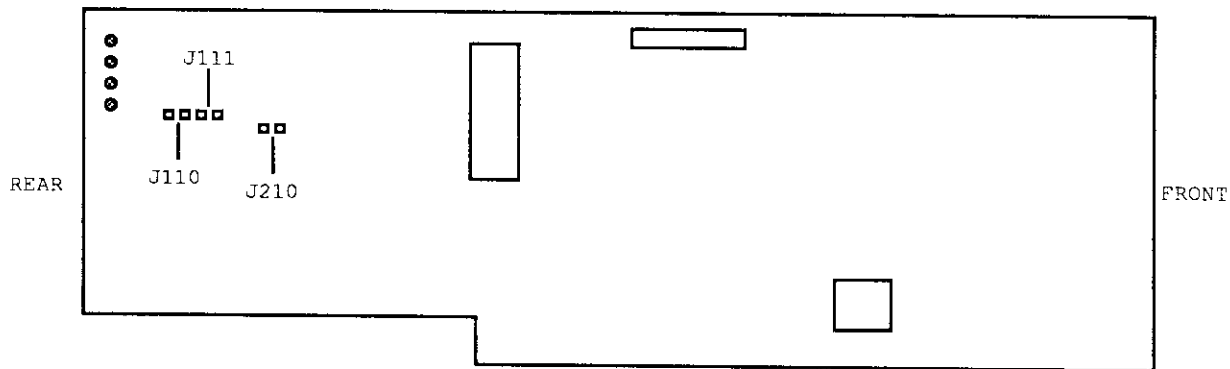


Figure 1-2. Location of the Diagnostic Options Jumpers on the A25/A28 Acquisition MPU Board

TABLE 1-2
Jumper Settings of the Acquisition Diagnostic Options Jumpers

<u>Operation Mode</u>	<u>J111</u>	<u>J210</u>	<u>J110</u>
<u>Normal</u> operation	Removed	Removed	Removed
Normal Operation With Calibration Bypassed	Installed	Removed	Removed
Power-up Arbitor Test Loop	Don't Care	Installed	Removed
Power-up Arbitor Test Loop With Timebase Interrupt	Don't Care	Installed	Installed (after power-up)

1.8 Equipment Required

The following equipment is recommended for use in troubleshooting the 11801, 11802, and SM-11 with the Diagnostic procedures presented in this section. For a complete list of all recommended maintenance equipment, refer to the 11801, 11802, and SM-11 Service Reference Manuals.

- TEKTRONIX 067-1267-00 Calibration Fixture: Troubleshooting Aid Extender Card with Cables.
- TEKTRONIX 067-1323-00 Acquisition System Extender.
- TEKTRONIX 067-1324-00 Acquisition Board Extender.
- TEKTRONIX P6401 Logic Probe.

- TEKTRONIX 1240 Logic Analyzer.
- TEKTRONIX 7904A Oscilloscope with 7A26 Dual Trace Amplifier, 7S14 Dual Trace Delayed Sweep Sampler, and 7B10 and 7B15 Time Bases, or equivalent.
- Test terminal, ANSI 3.64 compatible, with RS-232-C interface.

1.9 Circuit Coverage

The following list contains those circuits in the 11801/11802 Digital Sampling Oscilloscope mainframes and the SM-11 Multi-Channel Unit that are less than 50% functionally verified (directly or indirectly) by the Extended Diagnostics tests. Functional verification is 0% unless the item is classified as "partial", which indicates it is 1-50% tested. Numbers within "<>" indicate the associated schematic numbers. Circuits not on this list are 50 to 100% functionally tested by the Diagnostics.

To ensure good functional verification of the Timebase subsystem and interface boards to the Acquisition and sampling heads (i.e. Strobe/TDR Buffer, Head Interconnect, SM11 Strobe Drive, TDR Drive, etc.), sampling heads with TDR capability should be installed at one point or another in the testing process.

Input/Output (I/O) Board (A14)

- I/O-Internal & Control <21>
 - Real Time Decode/Request—partial
 - Realtime Clock—partial

Rear Panel Board (A12)

- Rear Panel—GPIB & RS-232-C <3>
 - GPIB Controller—partial
 - GPIB Data Buffers
 - GPIB Control Driver
 - GPIB Status

Strobe/TDR Buffer Board (A19)

- Strobe/TDR Buffer <15>
 - TDR buffer circuitry - can be verified with interactive test routines

Strobe Drive Board (A1)

- SD TDR Rate Generator <16B>
 - Internal Clock Output - can be verified with interactive test routines

Calibrator Assembly (A6) - can be verified with interactive test routines

Trigger Select Board (A30)

Trig Pickoff & Delay Line Comp. (A32) - 11802 only

Acquisition Analog Board (A24/A27)

- Acq Analog Programmable Gain Amp <9>
 - Chan Input Mux A (50%)
 - Chan Input Mux B (50%)
- Acq Analog Control Voltage Generation <10>
 - Outputs from head control function DACs to heads

2 Kernel Diagnostics

Kernel Diagnostics normally run at power-up to verify local processor hardware and to establish communication with other subsystems. A properly operating power supply is also necessary for these tests to run. If a mainframe subsystem is unable to run its Kernel Diagnostics (i.e. execute instructions from ROM), refer to Troubleshooting Non-Functioning Subsystem Kernels. Failures found during Kernel Diagnostics cause the failing subsystem to loop on the failed test and to produce an error index code on indicators within the instrument (and externally when possible) as discussed below.

Kernel Diagnostics begin by displaying "**Diagnostics in Progress**" and "**Comm Test in Progress**" on the screen. If the instrument is being powered up from a "cold" condition, the diagnostics may be complete before the CRT warms up enough to display these messages. Diagnostic routines are then performed in parallel on each of the instrument's processor subsystems: Executive, Display, Timebase, Mainframe Acquisition processors, and all attached SM-11 Multi-Channel Unit's Acquisition processors. Following successful execution of their kernel diagnostics, Acquisition processors attempt to communicate with the Timebase processor and the Timebase and Display processors attempt to communicate with the Executive processor.

Completion of Kernel Diagnostics is indicated by the start of Self-Test Diagnostics. This occurs even if the Executive is the only processor which has successfully passed its Kernel Diagnostics (although in the case where the Display processor has not communicated successfully with the Executive processor, the message indicating Self-Test Diagnostics are beginning will not appear on the screen). Failure of Kernel Diagnostics may be indicated by the message "**Dsy Kernel Failure**" or "**Comm Test in Progress**" remaining on the screen (for Display kernel failures), and/or a single high-low beep and illuminated menu button indicators (for Executive kernel failures). In addition, if either the Display processor, Timebase processor, or both Mainframe Acquisition processors do not successfully pass their communications stage, the instrument will automatically enter Extended Diagnostics at the end of Self-Test Diagnostics. (Of course, if the Display processor is at fault, the Extended Diagnostics menu will not appear on the screen.)

2.1 Error Index Overview

Kernel Diagnostics produce error index codes which point to test descriptions. The test descriptions, which are named for the circuits being tested, are found in the Test Descriptions portion of this manual. Since the condition of the instrument is unknown at power-up when a kernel failure occurs, Kernel Diagnostics in the Executive, Timebase, and Acquisition processors do not (and can not necessarily) display error index codes on the display screen. Instead, these processors generate hexadecimal (hex) numbers that are read as a series of binary bits from either internal test points or LED's. The Display Kernel Diagnostics produce an error message on the screen, if possible, giving the name of the test that failed. Each subsystem's Kernel Diagnostics generates error index codes in different ways and with different formats. The following sections will describe in detail the ways the error index codes are generated and how and where to read them for each subsystem.

The instrument will need some disassembly in order to see the status LEDs and to read the on-board status pins. Refer to the Service Reference Manual for cabinet and board removal information, including important warnings and cautions.

2.2 Test Status Indicators

All subsystem kernels contain two status LEDs that indicate the onset, failure, and completion of testing. The Executive processor contains five status pins from which an error index code can be read (via a logic probe or scope), while the Display processor contains eight status pins. The Executive and Display processors also have additional ways of displaying kernel test status which are discussed under their respective headings below. The Timebase and Acquisition processors use LEDs instead of status pins to indicate their error index codes (four for the Timebase and two for Acquisitions). Illustrations in the following sections show the locations of the kernel status LEDs and pins. These status indicators are used for other purposes during normal instrument operation, hence they are valid only while Diagnostics are running.

2.3 Start/Stop Trigger Pin or LED

When the Executive or Display processor is looping on a failed kernel test, the start/stop trigger pin transitions high at the start of the test and low at the completion. When the Timebase or an Acquisition processor is looping on a failed kernel test, the start/stop trigger LED transitions low (i.e. the "Test Running" LED is on) at the start of the test and high (i.e. the "Test Running" LED is off) at the completion. This signal may be useful for triggering test equipment during troubleshooting.

2.4 Executive Kernel

At power-up, two status LEDs on the A17 Main Processor board are turned on and five test status pins are set high (logic 1) by a hardware reset line (refer to the following tables and figure). When the Executive processor starts executing diagnostic code from its ROMs, it turns the forward LED off (LED 1). At the start of the second test, and of those that follow, the count on the status pins is decremented by one. Thus, a unique error code is registered on the status pins at the start of each test. When all Executive kernel tests pass, both LEDs are turned off and Executive Self-Test Diagnostics begin.

To gain access to the internal status pins and LEDs, remove the top cabinet cover and locate the A17 Main Processor board in the card-cage. The Service Reference Manual shows the location of most mainframe circuit boards. Use caution when working on the instrument with covers removed. The following figure shows the approximate location of the status LEDs and pins on the A17 Main Processor board. When reading an error index code with a logic probe or oscilloscope, carefully note the pin location for each bit. On the status pins, a TTL high is a logic one and a TTL low is a logic zero.

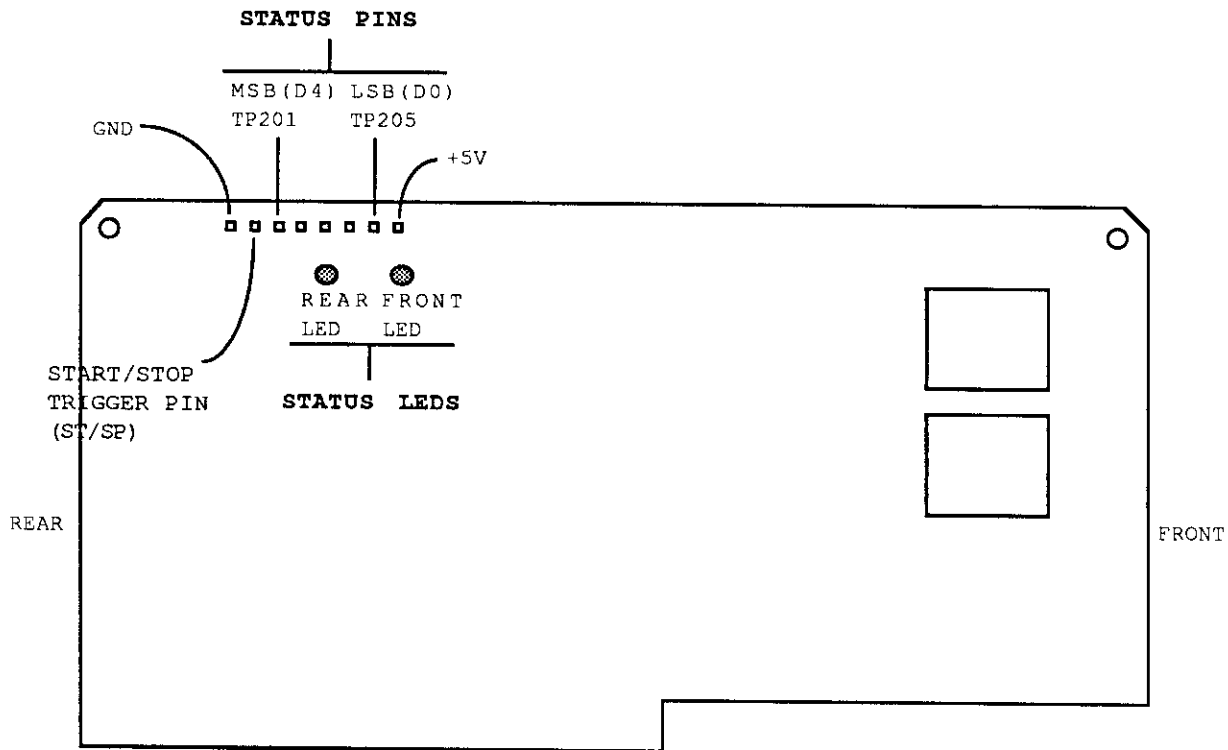


Figure 2-1. A17 Main Processor Board Status LEDs & Pins

If an Executive kernel test fails, but is able to finish execution, the rear (lit) LED will be turned off and the forward (off) LED will be turned on. If possible, a *single* high-low beep will be generated and the processor will begin looping on the failed test. To identify the failed test, the error index code can be read, via a logic probe or scope, on the Main Processor board status pins.

In addition to the internal status pins and LEDs, when signal paths to the front panel are good, the error index code for the failed test is registered on the front panel MENUS LEDs with "on" LEDs (logic 1) and "off" LEDs (logic 0). See Table 2-2 Executive Kernel Status Pins & MENUS LEDs Encoding below on how to create the error index code from the status pins or front panel LEDs. Also during Executive Kernel Diagnostics, the TOUCH PANEL ON/OFF and the ACQUISITION RUN/STOP front panel LEDs mirror the states of the Executive status LEDs on the Main Processor board, as indicated in Table 2-1 Executive Kernel Test Mode LEDs below.

If the Executive processor locks up because of erroneous execution of ROM instructions and is unable to finish running a test, the LEDs will not trade states and the processor will not be able to loop on the failed test. To determine whether the Executive processor is looping on a failure, locked-up, or did not start executing code from the ROMs, turn the ON/STANDBY switch to Standby, then back on and observe the status LEDs. Match the LED conditions to those in the following table to determine the fault condition.

TABLE 2-1
Executive Kernel Test Mode LEDs

	<u>A17 Rear Status LED</u> <u>(Front Panel On/Off LED)</u>	<u>A17 Forward Status LED</u> <u>(Acquisition Run/Stop LED)</u>
Power-on	On	On
Test Executing (or Lock-up)	On	Off
Test Failure (looping)	Off	On
Kernel Testing Completed	Off	Off

TABLE 2-2
Executive Kernel Status Pins & MENUS LEDs Encoding

<u>Low Hex Digit</u>	<u>MENUS Button LED</u>	<u>A17 Status Pin</u>	<u>Bit Number</u>
	WAVEFORM	TP205 (D0)	1 - LSB (least significant bit)
	TRIGGER	TP204 (D1)	2
	MEASURE	TP203 (D2)	3
	STORE/RECALL	TP202 (D3)	4
<u>High Hex Digit</u>	ENHANCED ACCURACY	TP201 (D4)	5 - MSB (most significant bit)

The Start/Stop Trigger Pin (ST/SP) goes high at the start of a kernel test and low at the end of the test. This can be used for test equipment (e.g. oscilloscope or logic analyzer) synchronization.

After all Executive kernel tests have run and passed, the Executive processor checks the Diagnostic Options Jumpers on the Input/Output Board. If the jumpers are set to loop on the RS-232 test, the Executive processor proceeds to do so. If the Diagnostic Options Jumpers are not set to loop on the RS-232 test, the Executive processor attempts to communicate with the other subsystems. The order of establishing subsystem communication is Display, then Timebase. If the Display or Timebase is unable to successfully communicate with the Executive processor, the Display or Timebase goes into an echo mode in which it inverts, then retransmits any data received from the Executive Subsystem. Regardless of communication problems with the other subsystems, the Executive processor will continue on to Self-Test Diagnostics.

2.4.1 Power-Up Options

Executive Kernel Diagnostics may be bypassed or forced to loop on the power-up RS-232 test with the Diagnostic Options Jumpers on the A14 Input/Output Board. The Kernel Diagnostics should be bypassed only in special circumstances because the operating condition of the instrument is unknown

when the Diagnostics are bypassed. See the discussion in the Diagnostic Options Jumpers section under Overview near the beginning of this manual.

2.5 Display Kernel

Display Kernel Diagnostics run at power-up as do kernel tests in other subsystems, but test failures are also written to the display as failure messages, if possible. Because the display may be defective, error index codes are also generated on the A7 Display Controller Board and can be read from status pins and LEDs (see figure and table below).

At power-up, two status LEDs on the A7 Display Controller board are turned on and eight test status pins are set high (logic 1) by a hardware reset line (refer to the following table and figure). When the Display processor starts executing diagnostic code from its ROMs, it turns the rear LED off (ST1), clears the display, and then writes "Diagnostics in Progress" to the display. At the start of the second test, and of those that follow, the count on the status pins is decremented by one. Thus, a unique error code is registered on the status pins at the start of each test.

To gain access to the internal status pins and LEDs, remove the top cabinet cover and locate the A7 Display Controller board lying just under the top cover. The Service Reference Manual shows the location of most mainframe circuit boards. Use caution when working on the instrument with covers removed. The following figure shows the approximate location of the status LEDs and pins on the A7 Display Controller board. When reading an error index code with a logic probe or oscilloscope, carefully note the pin location for each bit. On the status pins, a TTL high is a logic one and a TTL low is a logic zero. Pins 0 (LSB) to 3 make up the low hex digit and pins 4 to 7 (MSB) make up the high hex digit.

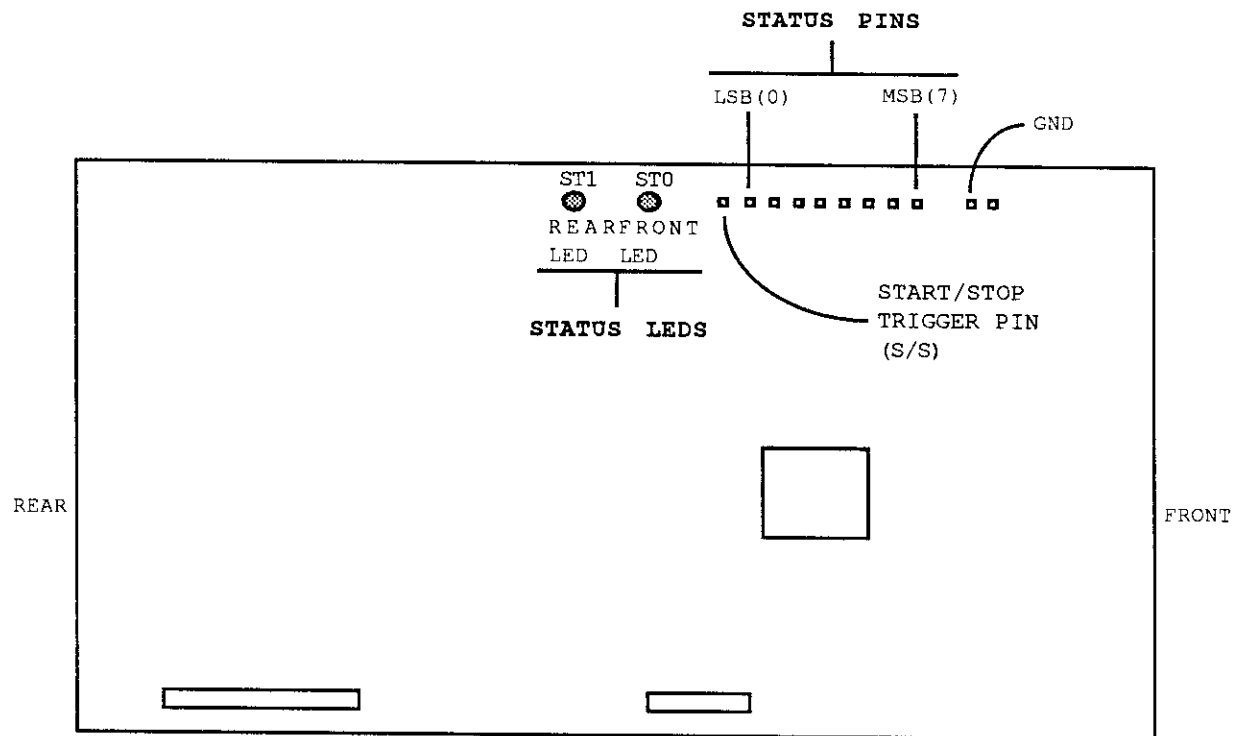


Figure 2-2. A7 Display Controller Board Status LEDs & Pins

If a Display kernel test fails, but is able to finish execution, the forward (on) LED will be turned off and the rear (off) LED will be turned on. If possible, a failure message such as

**Dsy Kernel Failure
RAM Data Bit**

will be written on the display and the processor will begin looping on the failed test. To identify the failed test (when a failure message does not appear on the display), the error index code can be read, via a logic probe or scope, on the Display Controller board status pins.

If the Display processor locks up because of erroneous execution of ROM instructions and is unable to finish running a test, the LEDs will not trade states and the processor will not be able to loop on the failed test. To determine whether the Display processor is looping on a failure, locked-up, or did not start executing code from the ROMs, turn the ON/STANDBY switch to Standby, then back on and observe the status LEDs. Match the LED conditions to those in the following table to determine the fault condition.

TABLE 2-3
Display Kernel Test Mode LEDs

	A7 Rear Status LED (ST1 LED)	A7 Forward Status LED (ST0 LED)
Power-on	On	On
Test Executing (or Lock-up)	Off	On
Test Failure (looping)	On	Off
Kernel Testing Completed	Off	Off

The Start/Stop Trigger Pin (S/S) goes high at the start of a kernel test and low at the end of the test. This can be used for test equipment (e.g. oscilloscope or logic analyzer) synchronization.

When the Display kernel tests have successfully completed and the communication test to the Executive processor is about to start, the Display processor writes the message "Comm Test in Progress" to the display. If communication is successful, the Display proceeds to Self-Test (assuming that the Diagnostic Options Jumpers are set to not bypass Self-Test). A communication failure causes the Display processor to enter an echo mode in which it inverts and sends back any received data from the Executive processor. This echo mode provides a means to help troubleshoot communication problems.

2.6 Timebase Kernel

At power-up, six status LEDs on the A5 Timebase Controller board are turned on by a hardware reset line (refer to the following table and figure). Two of the status LEDs (Test Running and Test Failure) are used to indicate the Timebase Kernel Test mode (see Table 2-4) and the other four (ST1 - ST4) are used to generate a single hexadecimal (4-bit) error index code. When the Timebase processor starts executing diagnostic code from its ROMs, it turns all of the status LEDs off. During the execution of each kernel test, the Test Running LED is turned on at the start of the test and off at the completion of each test. At the start of the first kernel test, the error index code on LEDs ST1 (LSB) through ST4 (MSB) is set to 01_{hex} (ST1 on, ST2 - ST4 off). At the start of each succeeding test, the error index code on LEDs ST1 - ST4 is incremented by one. Thus, a unique error code is registered on status LEDs ST1 - ST4 at the start of each test.

To gain access to the internal status LEDs, remove the top cabinet cover and locate the A5 Timebase Controller board lying under the bottom cover. The Service Reference Manual shows the location of most mainframe circuit boards. Use caution when working on the instrument with covers removed. The following figure shows the approximate location of the status LEDs on the A5 Timebase Controller board

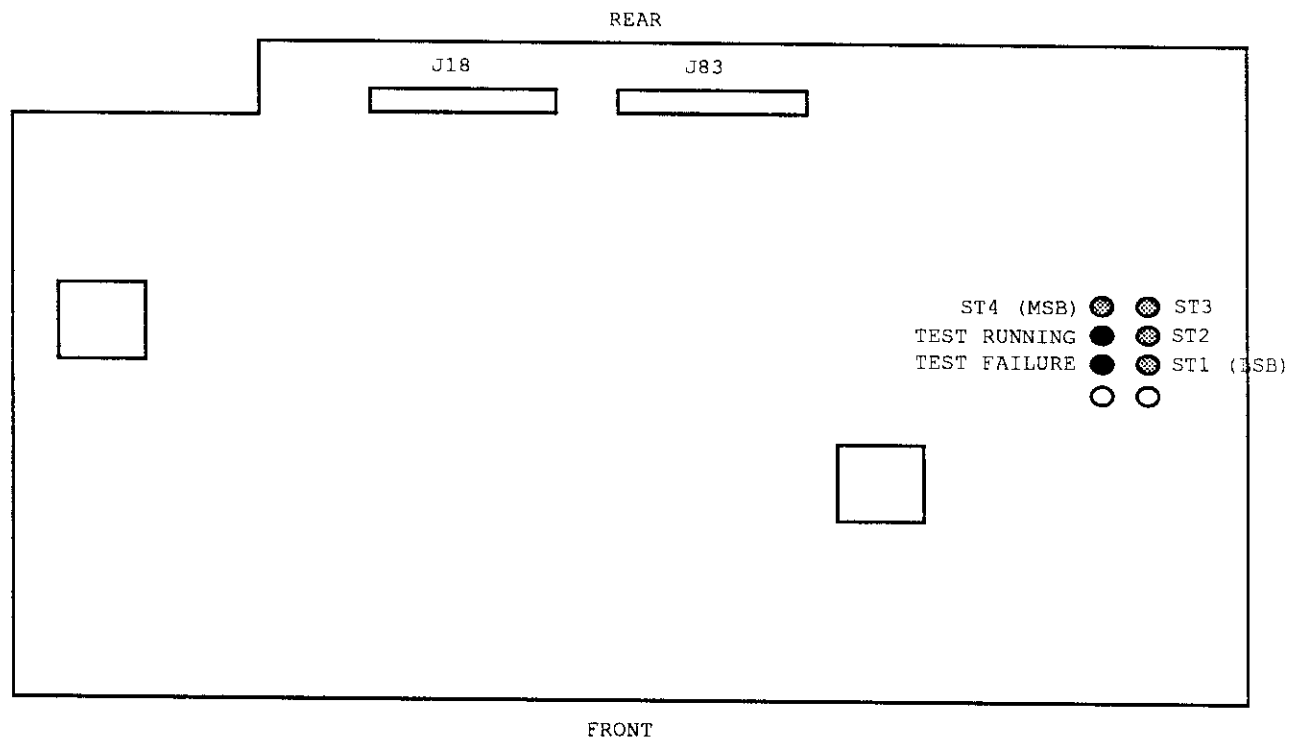


Figure 2-3. A5 Timebase Controller Board Status LEDs

If a Timebase kernel test fails, but is able to finish execution, the Test Failure LED will be turned on and the processor will begin looping on the failed test. At this point, the Test Running LED will also appear to be on (but maybe not as bright) since the processor is continually looping on the failed test. To identify the failed test, simply read status LEDs ST1 - ST4 from the Timebase Controller board and create the single hexadecimal (4-bit) error index code.

If the Timebase processor locks up because of erroneous execution of ROM instructions and is unable to finish running a test, the Test Failure LED will not turn on and the processor will not be able to loop on the failed test. To determine whether the Timebase processor is looping on a failure, locked-up, or did not start executing code from the ROMs, turn the ON/STANDBY switch to Standby, then back on and observe the status LEDs. Match the LED conditions to those in the following table to determine the fault condition. (Since the Power-on and Test Failure conditions are the same, you must observe the LEDs closely. If the LEDs blink or flicker before the two test mode LEDs both stay on, then it is likely that a Test Failure condition has occurred. Also, under a Test Failure condition, the Test Running LED is likely to be dimmer than the Test Failure LED or may actually be flickering off and on.)

TABLE 2-4
Timebase Kernel Test Mode LEDs

	<u>A5 Test Failure LED</u>	<u>A5 Test Running LED</u>
Power-on	On	On
Test Executing (or Lock-up)	Off	On
Test Failure (looping)	On	On or Flickering
Kernel Testing Completed	Off	Off

The signal which controls the Test Running LED can be used as a Start/Stop Trigger Pin similar to the Executive and Display. However, for the Timebase, this signal line goes low at the start of a kernel test and high at the end of the test (exactly opposite of the Executive and Display). This can be used for test equipment (e.g. oscilloscope or logic analyzer) synchronization.

After Timebase kernel tests have successfully completed, and just before the Timebase processor attempts to communicate with the Executive processor, the Timebase processor runs a test in order to detect and identify all Acquisition processors which are resident in the instrument (and in SM-11's for 11801's). When the Timebase completes this configuration step, a communication test with the Executive processor is attempted. If communication is successful, the Timebase proceeds to Self-Test (assuming that the Diagnostic Options Jumpers are set to not bypass Self-Test and that the Display processor successfully communicated with the Executive processor). A communication failure causes the Timebase processor to enter an echo mode in which it inverts and sends back any received data from the Executive processor. This echo mode provides a means to help troubleshoot communication problems.

2.6.1 Miscellaneous Non-Kernel LED Patterns

2.6.1.1 Diagnostic Operation

Hardware problems (and possibly some unknown firmware problems) which occur after Kernel Diagnostics have executed (and which may be intermittent or spurious) can cause the Timebase processor to display patterns on the Kernel LEDs (i.e. ST1 - ST4, Test Running, and Test Failure) which are not described in the Kernel Test Descriptions section. When one of the errors in the table below occurs during diagnostics operation, the Timebase processor displays the associated error code on the Kernel LEDs and continues to operate if possible. Continuing to operate may cause the error status to change or disappear and the system may behave strangely (it's unknown what that really means). The miscellaneous errors that the Timebase processor may recognize during diagnostics operation are:

TABLE 2-5
Timebase (Diagnostics Operation) Non-Kernel LED Patterns

<u>Suspected Error</u>	<u>A5 Test Failure LED</u>	<u>A5 Test Running LED</u>	<u>ST4 LED</u>	<u>ST3 LED</u>	<u>ST2 LED</u>	<u>ST1 LED</u>	
Unknown Interrupt	On	Off	On	Off	Off	On	(9)
Bad Message From Executive	On	Off	On	Off	On	On	(B)
Bad Message From Acquisition	On	Off	On	On	Off	Off	(C)
Bad Acquisition Interrupt	On	Off	On	On	Off	On	(D)

2.6.1.2 Normal Operation

If the Timebase processor, when operating in its normal (i.e. non-diagnostic) mode, detects certain invalid conditions (conceivably caused by various hardware or firmware problems), it displays the following error code on the Kernel LEDs and halts execution.

TABLE 2-6
Timebase (Normal Operation) Non-Kernel LED Patterns

<u>Suspected Error</u>	<u>A5 Test Failure LED</u>	<u>A5 Test Running LED</u>	<u>ST4 LED</u>	<u>ST3 LED</u>	<u>ST2 LED</u>	<u>ST1 LED</u>	
Any Fatal System Error	On	Off	On	On	On	On	(F)

2.7 Acquisition Kernel

At power-up, four status LEDs on the A25/A28 Acquisition MPU board are turned on by a hardware reset line (refer to the following table and figure). Two of the status LEDs (Test Running and Test Failure) are used to indicate the Acquisition Kernel Test mode (see Table 2-5) and the other two (DS100 and DS101) are used to generate a single 2-bit error index code. When the Acquisition processor starts executing diagnostic code from its ROMs, it turns all of the status LEDs off. During the execution of each kernel test, the Test Running LED is turned on at the start of the test and off at the completion of each test. At the start of the first kernel test, the error index code on LEDs DS100 (LSB) and DS101 (MSB) is set to 1 (DS100 on, DS101 off). At the start of each succeeding test, the error index code on LEDs DS100 and DS101 is incremented by one. Thus, a unique error code is registered on status LEDs DS100 and DS101 at the start of each test.

To gain access to the internal status LEDs of the 11801 or 11802 mainframe, the Acquisition system card cage must be removed and extended using the TEKTRONIX 067-1323-00 Acquisition System Extender. For SM-11 Multi-Channel Units, the SM11 Strobe Drive must be unscrewed and flipped backwards while the chassis plate underneath is pivoted up out of the way to reveal the status LEDs on the installed Acquisition MPU boards. Refer to the appropriate Service Reference Manual for procedures on how to accomplish these actions. Use caution when working on the instrument with covers removed. The following figure shows the approximate location of the status LEDs on the A25/A28 Acquisition MPU board.

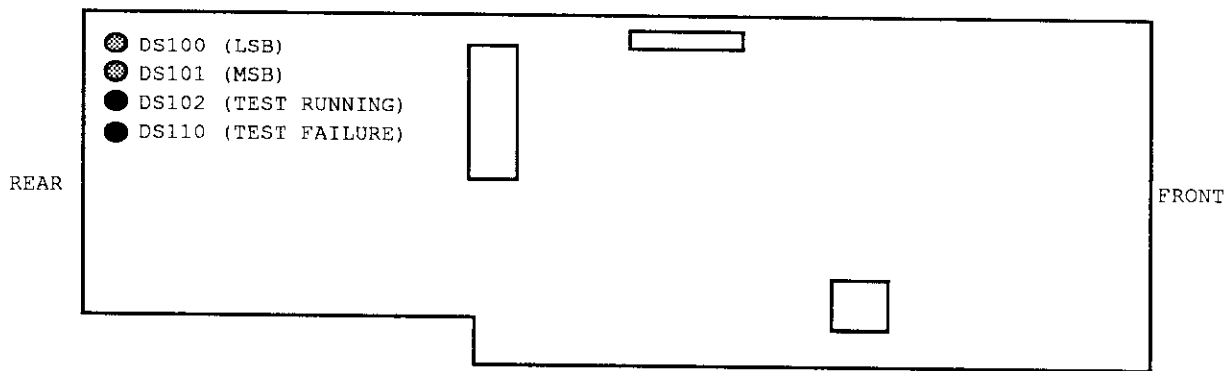


Figure 2-4. A25/A28 Acquisition MPU Board Status LEDs

If an Acquisition kernel test fails, but is able to finish execution, the Test Failure LED will be turned on and the processor will begin looping on the failed test. At this point, the Test Running LED will also appear to be on (but maybe not as bright) since the processor is continually looping on the failed test. To identify the failed test, simply read status LEDs DS100 and DS101 from the A25/A28 Acquisition MPU board and create the 2-bit error index code.

If the Acquisition processor locks up because of erroneous execution of ROM instructions and is unable to finish running a test, the Test Failure LED will not turn on and the processor will not be able to loop on the failed test. To determine whether the Acquisition processor is looping on a failure, locked-up, or did not start executing code from the ROMs, turn the ON/STANDBY switch to Standby, then back on and observe the status LEDs. Match the LED conditions to those in the following table to determine the fault condition. (Since the Power-on and Test Failure conditions are the same, you must observe the LEDs closely. If the LEDs blink or flicker before the two test mode LEDs both stay on, then it is likely that a Test Failure condition has occurred. Also, under a Test Failure condition, the Test Running LED is likely to be dimmer than the Test Failure LED or may actually be flickering off and on.)

TABLE 2-7
Acquisition Kernel Test Mode LEDs

	A25/A28 Test Failure LED	A25/A28 Test Running LED
Power-on	On	On
Test Executing (or Lock-up)	Off	On
Test Failure (looping)	On	On or Flickering
Kernel Testing Completed	Off	Off

The signal which controls the Test Running LED can be used as a Start/Stop Trigger Pin similar to the Executive and Display. However, for the Acquisition, this signal line goes low at the start of a kernel test and high at the end of the test (exactly opposite of the Executive and Display). This can be used for test equipment (e.g. oscilloscope or logic analyzer) synchronization.

After all Acquisition kernel tests have successfully completed, a communication test with the Timebase processor is attempted. If communication is successful, the Acquisition proceeds to Self-Test (assuming that the Diagnostic Options Jumpers are set to not bypass Self-Test and that the Executive communication paths to the Display and Timebase were successfully established). A communication failure causes the Acquisition processor to enter an echo mode in which it inverts and sends back any received data from the Timebase processor. This echo mode provides a means to help troubleshoot communication problems.

2.7.1 Power-Up Options

The Acquisition power-up sequence may be changed in order to help troubleshoot faults in the Acquisition-to-Timebase communication path. Diagnostic Options jumpers J110 and J210 on the A25/A28 Acquisition MPU Board (refer to the previous discussion on Diagnostics Options Jumpers for the Acquisition system) provide a way to invoke an interface arbiter test loop on power-up. This occurs after the Acquisition kernel tests have completed, but before it attempts the normal communications test with the Timebase processor. Refer to Troubleshooting A Failed Subsystem Communication Path for further information concerning the arbiter test loop.

2.7.2 Miscellaneous Non-Kernel LED Patterns

2.7.2.1 Diagnostic Operation

After the Acquisition kernel Timebase Communication test (refer to Acquisition Kernel Test Description 3), the Acquisition processor is supposed to receive a status word and associated interrupt from the Timebase processor to indicate whether the communication test patterns were received correctly by the Timebase processor. The status word the Acquisition processor receives from the Timebase processor normally indicates a "pass" condition (1234_{hex}) or a "fail" condition (FEDC_{hex}). If the status word indicates "pass", then the Acquisition processor continues on as normal. If the status word indicates "fail", the Acquisition processor goes back and loops on the Timebase Communication kernel test. However, if the Acquisition processor receives the Timebase interrupt and the status word indicates something other than the "pass" or "fail" condition (i.e. the status word is not 1234_{hex} or FEDC_{hex}), the Acquisition processor sees this as an unknown communication interrupt. The Acquisition processor then displays the following error code on the Kernel LEDs and loops forever, doing nothing.

TABLE 2-8
Acquisition (Diagnostics Operation) Non-Kernel LED Patterns

	A25/A28 Test Failure LED (DS110)	A25/A28 Test Running LED (DS102)	MSB LED (DS101)	LSB LED (DS100)	
<u>Suspected Error</u>					
Unknown Communication Interrupt	On	On	Off	Off	(0)

2.7.2.2 Normal Operation

If the Acquisition processor, when operating in its normal (i.e. non-diagnostic) mode, detects certain invalid conditions (conceivably caused by various hardware or firmware problems), it loops forever, doing nothing but displaying the following error code on the Kernel LEDs.

TABLE 2-9
Timebase (Normal Operation) Non-Kernel LED Patterns

<u>Suspected Error</u>	<u>A25/A28 Test Failure LED (DS110)</u>	<u>A25/A28 Test Running LED (DS102)</u>	<u>MSB LED (DS101)</u>	<u>LSB LED (DS100)</u>
Any Fatal System Error	Flashing	Flashing	Flashing	Flashing

2.8 Power-Up Subsystem Communications

Subsystem communication tests run after Kernel Diagnostics in order to verify communication lines between the Executive and Display, Executive and Timebase, and between the Timebase and Acquisition systems. Each subsystems' kernel communication test, except for the Executive, is documented as the last diagnostic routine in the appropriate Kernel Test Description section later in this manual.

2.8.1 Executive Subsystem Routines

The Executive portion of the subsystem communication tests are actually the first part of two high-level Self-Tests described in the MainFrm Comm area test descriptions for the Display (E561X) and Timebase (E562X). These routines are in the Executive Subsys Comm block. The communication routines have a "Subsystem Not Present" part followed by a "Subsystem Present" part. The "Not Present" part that runs at power-up is a low-level routine that sends and receives 16-bit words to determine if any communication is possible. If the Display and Timebase subsystems pass the first communications test, the second part of the MainFrm Comm tests (i.e., "Subsystem Present") is executed during Self-Test Diagnostics to verify high-level block data transfers. Refer to the test descriptions mentioned in this discussion for detailed descriptions of the communications tests.

2.8.2 Indications of Subsystem Communication Failures

If the Display or Timebase fails the first communications test on power-up, it will enter an echo mode in which it receives, inverts, and returns any data sent by the Executive. The internal status pins and/or LEDs for the failed subsystem will indicate that it is executing the failed Executive Comm test. However, the Start/Stop trigger pin or signal will not toggle. The Executive will proceed into Self-Test Diagnostics and label the failed subsystem as "Not Present", thus causing an Executive error to be generated. Review the Error Index description for the displayed Executive error index code to learn the actual failure mode. Also refer to the Display and Timebase Kernel Test Description sections for more information on Display and Timebase behavior when the Executive Communications test fails.

If the Display subsystem fails to communicate properly, it will also attempt to display the message:

**Dsy Kernel Failure
Exec Communication**

2.8.3 Troubleshooting A Failed Subsystem Communication Path

2.8.3.1 Executive to Display or Timebase

To troubleshoot a Subsys Comm test failure between the Executive processor and the Display or Timebase processor, select the Routine menu for the failed routine and set **Terse** and **Loop** "On" and **Stop on Err** "Off". Then select the failed routine and **Run**. Connect a logic analyzer to the data lines at the failed end and look for the series of data words listed in the Error Index portion of the test descriptions. Note that the first data pattern received by the Executive (after being inverted by the subsystem) that fails to match the expected pattern will cause the next pattern sent to the failing subsystem to be **DEAD_{hex}** and not the next pattern in the Error Index list. The **DEAD_{hex}** pattern is unexpected so the failing subsystem immediately enters the echo mode. The mainframe subsystem would also enter the echo mode automatically if the first pattern sent by the Executive was corrupted. After **DEAD_{hex}** is sent, the Executive resumes sending the data words from the Error Index list. Check the data path to the subsystem and the data path back to the Executive to locate the corrupted data bit(s) or control signal line.

2.8.3.2 Timebase to Acquisition

There are two ways to troubleshoot a communication failure between the Timebase processor and an Acquisition processor. The first is similar to troubleshooting Executive communication problems, as discussed in the previous section. This make use of the fact that the failing Acquisition unit should be looping on its power-up communications test with the Timebase (see Acquisition Kernel Test Description 3). Once the Extended Diagnostics menu appears on the display, the appropriate acquisition communications routine of the Timebase subsystem (see Timebase Self Test/Extended Test Description T311X) can be executed in a similar fashion as the Executive communication tests above (i.e. **Terse** and **Loop** "On" and **Stop on Err** "Off"). A logic analyzer or scope can then be used to view interface signals and busses.

The second method of troubleshooting Timebase-Acquisition interface problems uses special test routines, which exist in the Timebase and Acquisitions, that generate stimulus for the interface circuitry. The following steps must be taken for the Timebase and the failed Acquisition unit in order to make use of these special test routines.

For the failed Acquisition unit:

- Power down the instrument and extend the failed Acquisition unit MPU board by using the Acquisition System and Board Extenders (or the Acquisition Board Extender and extender cables for SM-11's, if possible). If the fault happens to lie in the Acquisition unit of an SM-11, it is possible to put that Acquisition in an extended mainframe Acquisition cage for troubleshooting.
- Place a jumper on J210 of the Acquisition MPU board. Refer to the sections on Acquisition Diagnostic Options Jumpers and Acquisition Power-up Options earlier in this manual for information on where this jumper is located. This jumper causes the Acquisition processor to enter a special arbitor test (see the Timebase M/F I/F Arbitor Test Acq 1 test description under the Timebase Self Test/Extended Test Descriptions) on power-up instead of executing the Timebase communication test (see Acquisition kernel test description 3).
- Power up the instrument and then place a jumper on J110. This enables Acquisition-to-Timebase interrupt generation, which must be done only after power-on reset, so that it can be checked.

For the Timebase:

- After the Extended Diagnostics menu appears on the display, the appropriate acquisition arbitor test routine of the Timebase subsystem can be executed. See the Timebase M/F I/F Arbitor Test Acq 1 test description for information on how the Timebase and Acquisition will behave when these special test routines are executing.

Once both the Acquisition and Timebase arbitor test exercisers are executing, a logic analyzer or scope can then used to view interface signals and busses.

3 Troubleshooting Non-Functioning Subsystem Kernels

This section describes how to troubleshoot the essential circuitry necessary to run the Kernel Diagnostics in each subsystem. The following procedures provide assistance in troubleshooting circuit faults that keep a kernel processor from running Kernel Diagnostic tests in on-board ROM. With this type of fault, the kernel is unable to give any indication of the fault location with the status LEDs or pins. Some of the following discussion may also help in troubleshooting a subsystem that is able to run Kernel Diagnostics. Refer to the earlier discussion for troubleshooting using the Kernel Diagnostics.

The troubleshooting procedures for all processors except the Executive involve setting jumpers in the kernel circuitry that isolate a processor from the rest of the system and force it to loop on a no-operation instruction. This stimulates the kernel circuitry by performing read operations throughout the processor's address range. Once the kernel circuit is setup, the test equipment recommended below is used, along with the Theory of Operation and schematics, to find the defective component. A special extender card is available for the Executive Kernel that raises it out of its card-cage for easy access and provides hardware stimulation for troubleshooting. The extender board can also be configured for use in troubleshooting the other card-cage boards.

Refer to the Service Reference Manual for information on board removal, component handling and replacement, adjustments required after board repair or battery replacement, and safety precautions. If circuit boards are repaired and need adjustment, refer to appropriate Checks and Adjustments sections in the appropriate Service Reference Manual.

Troubleshooting the Executive kernel is covered first, including a complete discussion on using the Kernel Extender Service Kit (067-1267-00), followed by information on troubleshooting the Display, Timebase, and Acquisition kernels.

3.1 Equipment Required

The following equipment or its equivalent is necessary when troubleshooting the 11801, 11802, or SM-11 subsystems' kernel circuitry.

- TEKTRONIX 067-1267-00 Calibration Fixture: Troubleshooting Aid Extender Card with Cables.
- TEKTRONIX 067-1323-00 Acquisition System Extender.
- TEKTRONIX 067-1324-00 Acquisition Board Extender.
- TEKTRONIX P6401 Logic Probe.
- TEKTRONIX 1240 Logic Analyzer.
- TEKTRONIX 7904A Oscilloscope with 7A26 Dual Trace Amplifier, 7S14 Dual Trace Delayed Sweep Sampler, and 7B10 and 7B15 Time Bases, or equivalent.

3.2 Executive Kernel

The Executive Kernel circuitry resides primarily on the Main Processor Board (A17), which plugs into the card-cage mother board. The Main Processor Board has no facilities to stimulate the kernel

circuitry, so a disabled Executive Kernel must be troubleshot with the Kernel Extender Service Kit. The service kit includes a special extender card that can be configured to exercise the kernel circuitry. The Main Processor Board has two jumpers and one connector that configure it to work with the extender card. Their settings are discussed under the Mode 1 and Mode 2 sections below. When troubleshooting the Executive Kernel, refer to the Theory of Operation and to schematic diagrams 27 and 28.

Read the Kernel Extender Card description below and follow the setup instructions to troubleshoot the Executive Kernel.

3.2.1 Kernel Extender Service Kit

The Kernel Extender Service Kit (067-1267-00) provides facilities for troubleshooting the Executive card-cage boards and the RS-232-C interface. Included in the kit are an extender card and the longer cables needed when using the the extender card. An RS-232-C loopback device is included in the kit to provide external feedback of test signals generated by Diagnostics. Use of the RS-232-C loopback device is discussed after the Extender Card.

The service kit contains the following:

Service Kit	067-1267-00
1 Extender/kernel troubleshooting board	380-8673-XX
6 Diagnostic Lead set cables	012-1144-00
1 50 pin 3M card-cage replacement cable	175-9919-00
1 50 pin 3M card-cage extender cable	175-9918-00
1 34 pin 3M card-cage replacement cable	179-0207-00
1 40 pin 3M card-cage replacement cable	179-0208-00
1 40 pin 3M card-cage extender cable	175-9917-00
1 8 pin/wire ribbon cable	174-0262-00
1 Edge card support guide	386-5418-00
1 RS-232-C Loopback Connector	013-0198-00
1 Ziplock circuit board pouch	

3.2.2 Using the Kernel Extender Card

The extender card raises the Executive card-cage boards above the card-cage to permit easy access for troubleshooting. It plugs into the Executive card-cage mother board to connect the system bus signals, which then become available on groups of test pins on the extender card, to the extended board. To use the extender card, remove the defective board and insert the extender board into the slot. Then plug the Executive board into the top of the extender card. For most boards, the overhead cables will need to be changed as discussed later under Overhead Cable Replacement.

The extender card can operate in either of two modes. The first is as a simple extender card that raises any of the Executive card-cage boards for troubleshooting with little or no performance degradation. The second mode allows the Executive kernel circuitry to be exercised for troubleshooting problems in which the Executive processor can not run its Kernel Diagnostics. Movable jumpers on the Extender card allow the card to be configured for either mode of operation. Procedures for configuring the Extender card and the Executive boards are given in the Mode 1 and Mode 2 sections below. The jumpers and their functions are as follows:

- J122 Used in conjunction with J320 to specify the number of wait-states. Jumper across pins 1-2 for one wait-state and across 2-3 for zero wait-states.

- J220 Jumper across pins 1 and 2 to latch the mother board addresses on connectors J100, J102, and J104.
- J320 Puts the extender card in the kernel stimulus mode when shorted across pins 1, 2, 3, and 4. Normally, the four-pin jumper is left on the two holding pins adjacent to pins 1 and 4. **Note:** *This mode should be used only when extending the Main Processor Board, and then, only when it is configured as described below.*

In either mode of operation, the state of any of the card-cage bus signals can be monitored on groups of test pins at the top left of the extender card. The diagnostic lead sets included in the service kit are designed for quick connection of these groups of test pins to a Tektronix 1240 logic analyzer. In addition, a jumper (J220) can be set to provide latched addresses on connectors J100, J102, and J104. The addresses are latched with the Executive kernel signal ALE. The pinouts for the test connectors are given later in this discussion.

3.2.2.1 Mode 1 -- Passive Extender

As stated previously, Mode 1 operates the extender card as a simple, passive extender card that raises any of the Executive card-cage boards for troubleshooting with little or no performance degradation. A set of extender or replacement cables is provided with the Diagnostic Service kit to allow extension of Executive cards that are connected to other cards through overhead cables. This mode can be used when the Executive Kernel Diagnostics has detected a fault or when troubleshooting with the Extended Diagnostics. To operate as a simple extender card:

1. Set the following extender card jumpers:
 - J320 Should be on the two holding pins adjacent to pins 1 and 4.
 - J122 Either position.
 - J220 Jumper pins 2 and 3 if latched addresses are desired on the extender card test pins (J100, J102, J104), otherwise jumper pins 1 and 2.
 - J120 Must not be connected to the Main Processor Board.
2. Insert extender card in place of card-cage board to be troubleshot.
3. Insert card-cage board into extender board and connect appropriate overhead-cable replacements from the service kit.

Replace jumpers to original positions when troubleshooting is completed.

3.2.2.2 Mode 2 -- Active Kernel Exerciser

In Mode 2, the extender card exercises the Executive kernel circuitry. If the kernel has a serious problem, then this mode exercises the support circuitry (e.g., address, data, and control lines to essential ROM) to help isolate the fault. This mode would typically be used only when none of the Kernel Diagnostic tests could run.

When both boards are properly configured and connected (see steps below), the extender card will perform sequential read operations throughout the Executive processor's address space. Address lines A1 through J120 are driven by a binary counter on the extender board. To get control of the 80286 buses the extender card asserts HOLD (J120-2). It then waits for the 80286 to tri-state its outputs and assert HLDA (J204-2). When HLDA is received, the extender card pulls S0(L) (J120-6) and COD/INTA (J120-7) low and generates S1(L) (J120-5) to perform memory reads. M/IO (J120-8) is held high to select the memory address space. The condition of the SRDY(L) signal line determines when S1(L) is generated. KREADADD (J120-4) is sent to the Main Processor Board to force the address buffers to reverse their drive direction while the extender card addresses are generated. The signals HLDA, SYS_CLK, SYS_RESET, and SRDY(L) that are generated by the Main Processor Board must be working for the extender card to function in this mode. To exercise the Executive Processor kernel:

1. Set the following extender card jumpers:

- | | |
|------|---|
| J320 | Jumper across pins 1 to 4 . |
| J220 | Jumper pins 1 and 2 if latched addresses are desired, otherwise jumper pins 2 and 3. |
| J122 | Jumper pins 2 and 3. This sets the number of wait states required to access Executive system ROM to zero. (Pins 1 and 2 select one wait state.) |
| J120 | Connect cable Tektronix part # 174-0262-00 which will later be connected to the Main Processor Board. |

2. Replace Main Processor Board with extender card.
3. Remove Main Processor Board jumper J660 to disable SRDY.
4. Insert Main Processor Board into extender board.
5. Remove Main Processor Board jumper on J800,pins 2-3 and connect the 8-wire ribbon cable from J120 on the extender card.
6. Connect appropriate overhead-cable replacements from the service kit.

Replace jumpers to original positions when kernel troubleshooting is completed.

3.2.2.3 Overhead Cable Replacement

Executive card-cage boards require different replacement cables for overhead cable connections. Table 3-1 defines the cables to use for extending each card-cage board. The cables are identified by their Tektronix part number.

TABLE 3-1
Overhead Cable Replacement Guide

Executive Board	Extend Cable	Replace Cable	Extended Cable	Replacement Cable
Main Processor		175-9814-00		179-0207-00
Input/Output	175-9915-00	175-9814-00 175-9854-00	175-9917-00	179-0207-00 179-0208-00
MMU	175-9808-00	175-9809-00	175-9918-00	175-9919-00
Compressor	175-9810-00	175-9809-00	175-9918-00	175-9919-00
Memory				
Memory Expansion				

3.2.2.4 Extender Card Test Points

Four test points allow the mother board power and ground lines to be probed. The test points are:

TP100 Ground
 TP200 +15 V
 TP204 -15 V
 TP300 Ground
 TP400 +5 V

3.2.2.5 Extender Card Connectors

Two types of test connectors with square pins are available on the extender card. All connectors but one are for probing the mother board address, data, and control signals. Pinouts for these test signal "J" connectors are given below. Connector J120 connects by cable to the Main Processor Board connector J800 when in the kernel stimulation mode (Mode 2). This connection should only be made when both boards are properly configured. See Mode 1 and Mode 2 jumper settings earlier in this discussion.

The "J" signal connector pin numbering is:

1	3	5	7
2	4	6	8

Table 3-2 gives an alphabetical listing of all signal names and their test-connector pin numbers. Connector pins not included in Table 3-2 have no connection to any mother board or extender board signal.

TABLE 3-2
Extender Board Test Signal Connector Pinout

Signal Name	Connector-Pin Number
A0	J302-1
A1	J302-2
A2	J302-3
A3	J302-4
A4	J302-5
A5	J302-6
A6	J302-7
A7	J302-8
A8	J308-1
A9	J308-2
A10	J308-3
A11	J308-4
A12	J308-5
A13	J308-6
A14	J308-7
A15	J308-8
A16	J200-1
A17	J200-2
A18	J200-3
A19	J200-4
A20	J200-5
A21	J200-6
A22	J200-7
A23	J200-8
A24	J206-1
A25	J206-2
A26	J206-3
A27	J206-4
A28	J206-5
A29	J206-6
A30	J206-7
A31	J206-8
ALE	J202-2
BHE(L)	J202-6
COD/INTA	J306-3
D0	J300-1
D1	J300-2
D2	J300-3
D3	J300-4
D4	J300-5
D5	J300-6
D6	J300-7
D7	J300-8
D8	J304-1
D9	J304-2

D10	J304-3
D11	J304-4
D12	J304-5
D13	J304-6
D14	J304-7
D15	J304-8
DEN	J202-7
DIAGNSIG(L)	J202-3
DT/R	J202-8
EPON	J204-1
HLDA	J204-2
INT0(L)	J204-5
INT1(L)	J204-3
INT2(L)	J106-1
INT3(L)	J106-2
INT4(L)	J106-3
INT5(L)	J106-4
INT6(L)	J106-5
INT7(L)	J106-6
INTA(L)	J202-1
IORC(L)	J204-7
IOWC(L)	J204-4
*LA1	J102-1
*LA2	J102-2
*LA3	J102-3
*LA4	J102-4
*LA5	J102-5
*LA6	J102-6
*LA7	J102-7
*LA8	J102-8
*LA9	J100-1
*LA10	J100-2
*LA11	J100-3
*LA12	J100-4
*LA13	J100-5
*LA14	J100-6
*LA15	J100-7
*LA16	J100-8
*LA17	J104-1
*LA18	J104-2
*LA19	J104-3
*LA20	J104-4
*LA21	J104-5
*LA22	J104-6
*LA23	J104-7
M/IO	J306-4
MRDC(L)	J202-4
MWTC(L)	J204-8
PCLK	J306-7
S0(L)	J306-6
S1(L)	J306-5
SRDY	J202-5

SYSCLK	J306-8
SYSRESET	J104-8

*Only active when J220 is shorted across pins 1 and 2.

3.2.3 Using the RS-232-C Loopback Device

The RS-232-C loopback device aids in troubleshooting the circuitry from the Executive Processor out to the standard RS-232-C port. Normally, the primary means of diagnostic test control is through the touch screen display. But, should the circuitry that controls the touch screen display be faulty, a test terminal can control the Diagnostics through the RS-232-C port. If the RS-232-C circuitry is also faulty, a hardware strap on the Input/Output Board (A14) can be set to force the diagnostics to continually exercise the circuitry from the Executive processor out to the loopback device. This provides a troubleshooting aid in establishing RS-232-C control of the diagnostics. The loopback test can also be invoked through the Extended Diagnostics menus.

To troubleshoot the RS-232-C system at power-up (using Kernel Diagnostics), power down the instrument and remove the Input/Output Board (A14). On the right-hand side of the board are the Diagnostics Options Jumpers labelled J710. The bottom two jumpers control Kernel Diagnostics operation and are usually both set to the right side. For this test, set the second one up from the bottom to the left. This will cause the instrument to run the kernel tests then loop on the RS-232 stimulus routine. Next, connect the loopback device on the back-panel RS-232-C port connection. When the instrument is powered up, the RS-232 loopback routine will run after all Kernel Diagnostics tests have run and passed. The loopback routine is described in the Executive Kernel Test Descriptions under the Std RS232 External Loopback (04) test. See the Diagnostic Options Jumpers section for further information on the options jumpers.

To run the test from Extended Diagnostics, invoke the RS232 Extern Loop routine from the RS232 area of the Input/Output block. See the RS232 Extern Loop (E455X) test description in the Executive Self-Test/Extended Test Descriptions.

3.3 Display Kernel

Display kernel circuitry consists of an 80186 microprocessor, clocks, ROM, RAM, and associated data and address buffers, decoders, latches, and busses. Display kernel circuitry resides on the A7 Display Controller board. Refer to the appropriate theory of operation discussion and the kernel schematic diagram for the Display <39> when troubleshooting the kernel circuitry.

Setting three jumpers on the Display Controller board puts the Display processor into a test mode in which it performs read cycles while looping through a certain address range. This mode is useful for identifying faults that prohibit the processor from executing instructions from EPROMs U602 and U612 <39>. When the kernel circuitry is defective, both diagnostic status LEDs on the Display Controller board will be lit at power-up (by a hardware reset) and remain on. Normally, one of the LEDs would go off when the first Kernel Diagnostic test started.

3.3.1 Setting Kernel Test Jumpers

The usual position of jumpers J5, J6, and J11 on the A7 Display Controller board are labelled NORM, while the test positions are labelled TEST. To enter the kernel test mode, power down the instrument, set these three jumpers to the TEST position, and then restore power to the instrument.

Jumper J6 disables the ROM data buffers that drive the processor data lines on read cycles. Jumpers J5 and J11 work in conjunction with pull-up resistors on the data lines to cause the processor to read *FDFDhex*, which it interprets as two "STD" (byte) instructions. This is essentially a "no-operation" instruction in that the processor does not perform any external bus activity. No other jumpers need to be moved in order to initiate kernel test looping.

There are several other signal points and jumpers that should be checked while the Display is in this special test mode. On the Display Controller board, check the following:

- TP30 - MPUCLK, 8 MHz square wave.
- TP39 - RESET, high (5 V).
- J7 - Should be in 256 position, jumper pins 1-2.
- J10 - Should be in NORM position, jumper pins 1-2.
- J12 - Should be in 512 position, jumper pins 1-2.
- J13 - Should be in 512 position, jumper pins 2-3.
- U612-1 - A16, eight pulses, 1 μ s apart with 4 ms between sets of pulses.
- U612-27 - A15, 2 MHz square wave.

3.3.2 Kernel Test Address Sequence

On power-up reset, the 80186 processor tries to read instructions from ROM, starting at address *FFFF0hex*. Instead of reading the ROM, the processor reads and executes the two "STD" instructions (*FDFDhex*) present on the data bus, then increments the address lines to the next word address (in this case *FFFF2hex*). This sequence of reading instructions from the data bus and then incrementing the address repeats indefinitely. The sequence of hexadecimal addresses that the 80186 processor will cycle through is as follows:

FFFF0	Power-Up Address
FFFF2	
FFFF4	
FFFF6	
FFFF8	
FFFFA	
FFFFC	
FFFFE	
00000	
00002	
00004	
00006	
•	
•	
•	
0FFEE	Jump to Power-Up Address

The lower 16 address lines increment through all values from 0000 - *FFFEhex*, while the upper four address lines go high only during eight accesses to the power-up reset vector (i.e., *FFFF0* - *FFFFEhex*). This non-linear address sequence is due to the internal programmable chip selects in the 80186 that assume default values on power-up. This causes some of the address range to be undefined and inaccessible.

If the processor does not cycle through the addresses, check clock and reset signals to the processor and check the data lines for the *FDFDhex* pattern. Connect an oscilloscope to the upper four address lines (only 2 are readily accessible on the boards) and use holdoff to find the pattern of eight, periodic positive pulses. The address sequence above can be verified with a Tektronix 1240 Logic Analyzer (see the following logic analyzer set-up).

3.3.3 Logic Analyzer Set-up

Address Lines:	MA0 - MA7 on U614 <39> MA8 - MA15 on U620 MA16 - MA17 on U526
Clock:	ALE(H) on U526 (rising edge) <39>

Note that since MA18 and MA19 are not readily accessible, the addresses acquired with the analyzer will be missing the two upper address bits. Hence, addresses such as *3FFF0hex* will be seen instead of *FFFF0hex*.

3.4 Timebase Kernel

Timebase kernel circuitry consists of an 80186 microprocessor, clocks, ROM, RAM, and associated data and address buffers, decoders, latches, and busses. Timebase kernel circuitry resides on the A5 Timebase Controller board. Refer to the appropriate theory of operation discussion and the kernel schematic diagram for the Timebase <17> when troubleshooting the kernel circuitry.

Setting one jumper on the Timebase Controller board puts the Timebase processor into a test mode in which it performs read cycles while looping through a certain address range. This mode is useful for identifying faults that prohibit the processor from executing instructions from EPROMs U400 and U410 <17>. When the kernel circuitry is defective, all diagnostic status LEDs on the Timebase Controller board will be lit at power-up (by a hardware reset) and remain on. Normally, some of the LEDs would go off when the first Kernel Diagnostic test started.

3.4.1 Setting the Kernel Test Jumper

The usual position of jumper J730 on the A5 Timebase Controller board is labelled NORM, while the test position is labelled TEST. To enter the kernel test mode, power down the instrument, set this jumper to the TEST position, and then restore power to the instrument.

Jumper J730 disables the ROM data buffers that drive the processor data lines on read cycles. This jumper also forces processor data lines 1 and 9 low, via U741A and U741B, to cause the processor to read *FDFDhex*, which it interprets as two "STD" (byte) instructions. This is essentially a "no-operation" instruction in that the processor does not perform any external bus activity. No other jumpers need to be moved in order to initiate kernel test looping.

There are two other signal points that should be checked while the Timebase is in this special test mode. On the Timebase Controller Board, check the following:

- TP590 - RESET, high (5 V).
- U250-14 - CLKOUT, 8 MHz square wave.

3.4.2 Kernel Test Address Sequence

On power-up reset, the 80186 processor tries to read instructions from ROM, starting at address $FFFF0_{hex}$. Instead of reading the ROM, the processor reads and executes the two "STD" instructions ($FDFD_{hex}$) present on the data bus, then increments the address lines to the next word address (in this case $FFFF2_{hex}$). This sequence of reading instructions from the data bus and then incrementing the address repeats indefinitely. The sequence of hexadecimal addresses that the 80186 processor will cycle through is as follows:

FFFF0	Power-Up Address
FFFF2	
FFFF4	
FFFF6	
FFFF8	
FFFFA	
FFFFC	
FFFFE	
00000	
00002	
00004	
00006	
•	
•	
•	
0FFEE	Jump to Power-Up Address

The lower 16 address lines increment through all values from $0000 - FFFE_{hex}$, while the upper four address lines go high only during eight accesses to the power-up reset vector (i.e., $FFFF0 - FFFFE_{hex}$). This non-linear address sequence is due to the internal programmable chip selects in the 80186 that assume default values on power-up. This causes some of the address range to be undefined and inaccessible.

If the processor does not cycle through the addresses, check clock and reset signals to the processor and check the data lines for the $FDFD_{hex}$ pattern. The address sequence above can be verified with a Tektronix 1240 Logic Analyzer (see the following logic analyzer set-up).

3.4.3 Logic Analyzer Set-up

Address Lines	MA0 - MA7 on U700 <17> MA8 - MA15 on U710 MA16 - MA19 on U810
Clock:	ALE(H) on TP554 (rising edge) <17>

3.5 Acquisition Kernel

Acquisition kernel circuitry consists of a 6809 microprocessor, clocks, ROM, RAM, and associated data and address buffers, decoders, and busses. Acquisition kernel circuitry resides on the A25/A28 Acquisition MPU board. Refer to the appropriate theory of operation discussion and the kernel schematic diagram for the Acquisition <12> when troubleshooting the kernel circuitry.

To troubleshoot Acquisition kernel problems, the Acquisition System and Board Extenders may need to be used to provide access to the Acquisition MPU board.

Setting three jumpers on the Acquisition MPU board puts the Acquisition processor into a test mode in which it performs read cycles while looping through its full address range. This mode is useful for identifying faults that prohibit the processor from executing instructions from EPROM U611 <12>. When the kernel circuitry is defective, all diagnostic status LEDs on the Acquisition MPU board will be lit at power-up (by a hardware reset) and remain on. Normally, some of the LEDs would go off when the first Kernel Diagnostic test started.

3.5.1 Setting Kernel Test Jumpers

Normally, the pins of J500 on the A25/A28 Acquisition MPU board are shorted together by a resident jumper, while J501 and J502 are open (i.e. no jumper). To enter the kernel test mode, power down the instrument, remove jumper J500, install jumpers on J501 and J502, and then restore power to the instrument.

Jumper J500 disables the EPROM data buffer that drives the processor data lines on read cycles. Jumpers J501 and J502 work in conjunction with pull-up resistors on the data lines to cause the processor to read $5F_{hex}$, which it interprets as a "CLRB" (byte) instruction. This is essentially a "no-operation" instruction in that the processor does not perform any external bus activity. No other jumpers need to be moved in order to initiate kernel test looping.

There are a few signal points that should be checked while the Acquisition is in this special test mode. On the Acquisition MPU board, check the following:

- U830B-6 - RESET(L), high (5 V).
- TP1000 - 6.67 MHz rectangular wave (50 ns up, 100 ns down).
- TP420 - EN(L) 1.67 MHz square wave.

3.5.2 Kernel Test Address Sequence

On power-up reset, the 6809 processor tries to read a start-up vector (to which it will jump to execute instructions) from ROM at address $FFFE_{hex}$ and $FFFF_{hex}$. Instead of reading the ROM, the processor reads two "CLRB" bytes from the data bus and jumps to that address ($5F5F_{hex}$) to start executing instructions. It executes the "CLRB" ($5F_{hex}$) instruction and then increments the address lines to the next address (in this case $5F60_{hex}$). This sequence of reading instructions from the data bus and then incrementing the address repeats indefinitely. After it executes the "CLRB" instruction at address $FFFF_{hex}$, the address wraps around to 0000_{hex} and starts executing "CLRB" instructions from there. The total power-up sequence of hexadecimal addresses that the 6809 processor will cycle through is as follows:

FFFE	Power-Up Address
FFFF	
5F5F	
5F60	
5F61	
5F62	
5F63	
•	
•	
•	
FFFF	
0001	Low Memory Address
0002	
0003	
0004	
•	
•	
•	
FFFF	Jump to Low Memory Address

Therefore, after the initial power-up sequence is completed, the 16 address lines increment repeatedly through all values from 0000 - FFFF_{hex}.

If the processor does not cycle through the addresses, check clock and reset signals to the processor and check the data lines for the 5F_{hex} pattern. The address sequence above can be verified with a Tektronix 1240 Logic Analyzer (see the following logic analyzer set-up).

3.5.3 Logic Analyzer Set-up

Address Lines	A0 - A7 on U410 <12> A8 - A15 on U510
Clock:	EN(L) on TP420 (rising edge) <12>

4 Self-Test Diagnostics

As stated previously in the Overview section, Self-Test Diagnostics run automatically on power-up or when invoked by the user from normal operation. When Self-Test begins, the message **Self-Test in Progress** is shown on the display. During the majority of time that Self-Test runs, subsystems perform independent testing in parallel.

If any Self-Test failures occur, the instrument emits a *double* high-low beep. Error index codes generated from Self-Test failures are identical to the error index codes generated from the Extended Diagnostics. See the next section for further information on how the error index codes are generated and what they mean.

NOTE

Pressing front panel buttons, turning knobs, or touching the Touch Panel while Self-Test Diagnostics are running may cause a diagnostic failure.

5 Extended Diagnostics

Extended Diagnostics provides the means for examining and exercising individual tests or groups of tests, primarily for troubleshooting purposes, but also as a way of increasing the user's confidence level of instrument functionality.

A Self-Test failure or selecting Extended Diagnostics from the Utility menu produces a menu display showing all major instrument subsystems. When displayed as a result of a Self-Test failure, the status of each subsystem will be shown next to its name. When Extended Diagnostics are invoked from the Front Panel, no tests run initially, hence there is no test status displayed.

5.1 Error Index Overview

The format of the error index codes is based on the Extended Diagnostics menu structure. The Extended Diagnostics menus (described in detail later in this section) are in a four-level hierarchy with the Subsys(tem) menu at the highest level. Each major subsystem name in the Subsystem menu can be individually selected and tested. A selected Subsystem is broken into a number of circuit blocks in the Block menu, the second level. Touching the Block label at the bottom of the menu displays the Block Menu for the selected subsystem. Circuit blocks are then broken into a number of circuit areas in the Area menu, the third level. Touching the Area label at the bottom of the menu displays the Area Menu for the selected block. Each circuit area has a Routine menu, the fourth and last level, associated with it that has one or more selectable routines. Routines are the smallest test unit that can be selected and run. This Subsystem, Block, Area, and Routine menu hierarchy is used in generating the error index codes and organizing the test descriptions.

Extended Diagnostics error index codes are five digit codes whose first character indicates the subsystem tested. The last four digits are hexadecimal (hex) numbers that indicate the block, area, routine, and specific failure mode. For example, E3321 is decoded as follows:

E	Subsystem - Executive
3	Block name - Front Panel
3	Area name - Soft Keys
2	Routine name - Column Open
1	Failure Identity - specific failure mode

The subsystem character of an error index code will be one of the following:

E	Executive Subsystem
D	Display Subsystem
T	Timebase Subsystem
m	Mainframe Acquisition Subsystems
a	Multi-Channel Unit A Acquisition Subsystems
b	Multi-Channel Unit B Acquisition Subsystems
c	Multi-Channel Unit C Acquisition Subsystems
d	Multi-Channel Unit D Acquisition Subsystems

The failure mode digit (i.e., the right-most digit) of an error index code points to a discussion of a particular type of circuit failure. Failure mode discussions are located at the end of each test

description in numerical order. Each test routine can have up to 15 (1-*Fhex*) failure modes and associated discussions.

Self-Test/Extended Diagnostics error index codes refer to test descriptions which are divided into subsystem groups under the headings Executive (E), Display (D), etc., as shown above. However, since Multi-Channel Unit Acquisition tests are identical to Mainframe Acquisition tests, all of these test descriptions are described together in one section later in this manual. The test descriptions under each subsystem heading are in numerical order. For instance, the Executive subsystem test descriptions start with test number E111X. Next in order is E112X, E113X, ..., E571X. Likewise, the Display tests start with D111X and the Timebase tests with T111X. This order of subsystem headings and the numeric order of test descriptions reflects the actual test sequence for that particular subsystem.

5.2 Extended Diagnostics Menu Displays

The Subsys(tem) Menu is the top level of a four-level menu structure. In the second-level Block Menu, the selected subsystem is divided into circuit blocks. In the third-level Area Menu, the selected circuit block is divided into circuit areas. The Routine Menu, which is the lowest level, shows the routines that make up the selected area. Routines are the smallest test unit that can be individually run. Figure 5-1 shows an example of the top-level Subsystem Menu. The menu features discussed here can also appear on a remote terminal, when the Extended Diagnostics are used in the front panel emulation mode.

11801, 11802, & SM-11 Diagnostics

Figure 5-1. Subsystem Menu Example

Menu items are selected or started by touching their labels on the screen, just as in normal instrument operation. For example, to view the Block menu for a selected subsystem, touch the (2) Block label in the control section at the bottom of the Subsystem menu. Most menu items are highlighted when selected. When (r) Run is selected, the highlighted test will be performed. Touching (q) Quit, any front panel button, or any point on the screen will halt testing after the current routine finishes.

The characters and numbers in parenthesis, preceding the selectable menu items, are the single key commands used on a remote terminal in the front panel emulation mode. Their use is covered in the Remote Diagnostics discussion later in this manual.

Diagnostic menus are divided into three functional sections. The top two lines are for prompt and warning messages. The remainder of the menu except the bottom fourth is used for test status information. The bottom section of the menu is the test control area. The common features and functions of each menu section will be discussed in the following Subsystem menu description. Then, the features unique to the Block, Area, and Routine menus will be discussed.

5.2.1 Subsys(tem) Menu

The Subsystem Menu (see Figure 5-1) is the highest level of the Extended Diagnostics menu structure. It shows the status of all tests at a glance for each major subsystem of the instrument.

5.2.1.1 Prompt/Warning Section

The two top menu lines display messages that prompt you to do something or warn you about a problem in operation. For instance, pressing the HARDCOPY button when no printer is connected or when the printer cannot print (e.g., off line, no paper, etc.) will cause the following warning to be displayed:

Hardcopy absent or off line.

Warning or error messages are displayed as required.

Warning or error messages (those that are shown as selectable or highlighted) can be removed by touching the screen. Other prompt-type messages (those that are not shown as selectable), which are associated with a selected routine, can only be removed by selecting a different diagnostic test.

5.2.1.2 Test Status Section

The center section of the display has four columns, three of which are labelled SUBSYSTEM, INDEX, and FAULTS, that provide the following information:

SUBSYSTEM	This column shows the major instrument subsystems in the order in which they are tested. Selecting an entry causes its name to appear below the (1) Subsys label in the control section of the menu. Selecting (r) Run would execute all of the tests in this subsystem.
INDEX	This column gives the status of each subsystem as one of the following: pass The test has run and had no failure.

PXXXX	The test has failed with this error index code, which is a pointer to a test description. See the Error Index Overview discussion earlier in this section.						
*****	The test has not yet been executed, but when it is run, will give pass/fail status.						
-----	An interactive test requires some setup (e.g., install the RS-232 loop-back fixture), but when it is run, will give pass/fail status..						
(blank)	The test requires some interaction with the instrument such as that required for the Front Panel Soft Keys test. No pass/fail status is generated when run.						
?????	The hardware option indicated by the menu name was not installed (or the communication path(s) to a subsystem was not working).						
FAULTS	This column gives the total number of failures that have occurred in the subsystem. The count is increased by one for each failed routine. Only test routines that have run and failed will increment the fault count.						
Execution Mark	This column is a one-character field within the SUBSYSTEM column, preceding the selector name and succeeding the keystroke selector identifier. The symbol in this one-character field reveals the following status information for the routines underneath the particular selector: <table> <tr> <td>" "</td><td>All routines under this selector are available for execution when (r) Run is invoked.</td></tr> <tr> <td>"*"</td><td>No routines under this selector are available for execution when (r) Run is invoked.</td></tr> <tr> <td>"-"</td><td>One or more, but not all, routines under this selector are available for execution when (r) Run is invoked.</td></tr> </table> <p>Refer to the (-) Delete and (+) Add label descriptions in the Control Section below for further information concerning <i>execution marks</i>.</p>	" "	All routines under this selector are available for execution when (r) Run is invoked.	"*"	No routines under this selector are available for execution when (r) Run is invoked.	"-"	One or more, but not all, routines under this selector are available for execution when (r) Run is invoked.
" "	All routines under this selector are available for execution when (r) Run is invoked.						
"*"	No routines under this selector are available for execution when (r) Run is invoked.						
"-"	One or more, but not all, routines under this selector are available for execution when (r) Run is invoked.						

5.2.1.3 Control Section

The bottom fourth of the menu provides a number of functions for controlling which test(s) will run and the operating mode while the test is running. Also, there are selectors for moving between different menu levels and exiting the diagnostics altogether.

The control functions and their definitions are as follows:

- | | |
|------------|--|
| (1) Subsys | Selects the Subsystem Menu for display. Shows the currently selected instrument subsystem. |
| (2) Block | Selects the Block Menu for display. Shows the currently selected circuit block. |

- (3) **Area** Selects the Area Menu for display. Shows the currently selected circuit area or the first area to have a failure in the block.
- (4) **Routine** Selects the Routine Menu for display. Shows the currently selected routine or the first routine to fail in the block/area.
- (?) **Help** Displays a list of Field Replaceable Units (FRU's) for the currently selected routine. If the routine had a failure, then the error index code for that routine is also displayed. The FRU's are arranged as the most-to-least probable cause for errors in the routine. See the Help Menu section later in this manual for further information on the menu that this function displays.
- (-) **Delete** Places an *execution mark* on the currently selected menu name (in the upper menu area) and all items beneath it, clear down to the Routine level. This inhibits the actual execution of these routines, when (r) **Run** is invoked.
- Visually, *execution marks* are displayed on the screen as an asterisk (**) immediately to the left of the menu name (in the upper menu area). If one, but not all, menu names in a menu are marked with an asterisk, then the menu name for that menu at the next highest level (i.e. from Area to Block) will show an *execution mark* of a minus ('-').
- (+) **Add** Removes the *execution mark*, if present, on the currently selected menu name and all *execution marks* on items (i.e. blocks, areas, or routines) below the currently selected menu name (clear down to the Routine level). The *execution mark*, if present, will change from an asterisk (**) or minus ('-') to a space (' ').
- (D) **Debugger** This function, normally not selectable, is for use by qualified service personnel only. See the Debugger Menu section later in this manual for further information concerning the access and use of the menu that this function displays.
- (E) **Exit** Causes a confirmation prompt to appear in the Prompt/Warning area of the display. The second consecutive invocation of this selector causes the Extended Diagnostics mode to terminate and return the instrument to normal operation. Faults that disable the instrument will cause Exit to be non-selectable.
- (p) **Loop** Toggles On and Off. When On, the selected test(s) run in a loop continuously until stopped. The status shows the current number of completed loops when (t) **Terse** is Off. When (t) **Terse** is On, the count remains at zero until the test is stopped, then the number of completed loops is displayed. Running more than 65534 loops cause the overflow status, 65535+. The Loop status is set to zero when (r) **Run** is selected. When set to Off, a selected test that is run will be performed once only.
- (t) **Terse** Toggles On and Off. When On, the display status is not updated while tests are running. When testing stops, all screen status is updated. This provides the fastest possible routine execution for troubleshooting. When Off, status is updated as tests are executed.
- (x) **All** Toggles On and Off. When On, all tests in the current menu are selected for execution. When Off, only the selected test in the current menu will execute.

- (s) **Stop on Err** Toggles On and Off. When On, testing stops as soon as the first failure occurs. When Off, all selected tests will run to completion or continue to loop if (p) **Loop** is On.

NOTE

*If (s) **Stop on Err** and (p) **Loop** are both On and a failure occurs, the speaker will produce two high-low beeps (similar to the Self-Test indication). This allows unattended operation while waiting for intermittent failures to occur. This feature applies to all four diagnostic menu levels (i.e. Subsystem, Block, Area, and Routine).*

- (r) **Run/(q) Quit** Starts the currently selected test(s) when (r) **Run** is selected and changes the status from Stopped to Running. While running tests, the control label changes to (q) **Quit**. Touching (q) **Quit** stops testing at completion of the current routine and changes the status to Stopped. Testing will also stop when any point on the screen is touched or a front panel button is pressed.

5.2.1.4 Front Panel Buttons & Knobs

Three front panel buttons and both knobs are active during the Subsystem Menu (in addition to the Block, Area, and Routine Menus). They provide the following functions:

Touch Panel On/Off	Disables/enables alternately the touch panel from responding to user touches.
Hardcopy	Produces a hardcopy of the screen on an instrument-compatible printer. Refer to the section on Hardcopy for details.
Waveform	When used in the proper sequence with the Touch Panel On/Off button, enables the (D) Debugger selector. See the Debugger Section for the proper sequence of button presses.
Knobs	Controls the screen intensity.

5.2.2 Block Menu

The Block Menu (see Figure 5-2) is the second level of the Extended Diagnostics menu structure. Touching the (2) **Block** selector at the bottom of the screen displays the Block Menu for the currently selected subsystem. The Block Menu's Prompt/Warning Section and Control Section operate just as they do for the Subsystem Menu. Refer to the Subsystem Menu description for more information.

43

Figure 5-2. Block Menu Example

5.2.2.1 Test Status Section

The center section of the display has four columns, three of which are labelled BLOCK, INDEX, and FAULTS, that provide the following information:

BLOCK	This column shows the major circuit blocks in the order in which they are executed when (r) Run is invoked. Selecting an entry causes its name to appear below the (2) Block label in the control section of the menu. Selecting (r) Run would execute this test block.
INDEX	This column gives the status of each test area. The status types are the same as those in the Subsystem Menu. Refer to the Subsystem Menu discussion.
FAULTS	This column gives the total number of failures that have occurred in the circuit block. The count is increased by one for each failed routine. Only test routines that have run and failed will increment the fault count.
Execution Mark	This one-character field operates the same as for the Subsystem Menu except that the <i>execution mark</i> refers to routines in the particular block.

5.2.3 Area Menu

The Area Menu (see Figure 5-3) is the third level of the Extended Diagnostics menu structure. Touching the (3) Area selector at the bottom of the screen displays the Area Menu for the currently selected block. The Area Menu's Prompt/Warning Section and Control Section operate just as they do for the Subsystem Menu. Refer to the Subsystem Menu description for more information.

EXTENDED DIAGNOSTICS

AREA	INDEX	FAULTS
a) Control	pass	
b) Hard Keys	E3211	1
c) Soft Keys	pass	
d) *Knobs	****	
e) Verify		

(1) Subsys	(2) Block	(3) Area	(4) Routine
Executive	Front Panel	Hard Keys	Open
(2) Help	(-) Delete	(+) Add	(D) Debugger
FRU	Test (s)	Test (s)	Low-Level
(P) Loop	(t) Test	(R) All	(S) Stop on Err
Off	Off	Off	Off
			Stopped

Figure 5-3. Area Menu Example

5.2.3.1 Test Status Section

The center section of the display has four columns, three of which are labelled AREA, INDEX, and FAULTS, that provide the following information:

AREA	This column shows the circuit areas for the selected block in the order in which they are executed when (r) Run is invoked. Selecting an entry causes its name to appear below the (2) Area label in the control section of the menu. Selecting (r) Run would execute this test area.
INDEX	This column gives the status of each test area. The status types are the same as those in the Subsystem Menu. Refer to the Subsystem Menu discussion.
FAULTS	This column gives the total number of failures that have occurred in the circuit area. The count is increased by one for each failed routine. Only test routines that have run and failed will increment the fault count.
Execution Mark	This one-character field operates the same as for the Subsystem Menu except that the <i>execution mark</i> refers to routines in the particular block.

5.2.4 Routine Menu

The Routine Menu (see Figure 5-4) is the fourth and lowest level of the Extended Diagnostics menu structure. Touching the (4) **Routine** selector at the bottom of the screen displays the Routine Menu for the currently selected area. The Routine Menu's Prompt/Warning Section operates just as it does for the Subsystem Menu. Refer to the Subsystem Menu description for more information.

EXTENDED DIAGNOSTICS

ROUTINE	INDEX	FAULTS	ADDRESS	EXPECT	ACTUAL
a) Open	E3211	1	003300	03FF	01FF
<div style="border: 1px solid black; height: 400px; width: 100%;"></div>					

Figure 5-4. Routine Menu Example

5.2.4.1 Test Status Section

The center section of the display has seven columns, three of which are labelled ROUTINE, INDEX, and, FAULT. Three others are test result columns giving either the test address, expected data, and actual data; or the acceptable data minimum, acceptable data maximum, and the actual data. This low-level information can be useful for troubleshooting the failure. The columns provide the following information:

ROUTINE	Shows the routines for the selected area in the order in which they are executed when (r) Run is invoked. Additionally, some interactive routines may be listed with their Index status shown as "----" or blank. Selecting an entry causes its name to appear below the (3) Routine label in the control section of the menu. Selecting (r) Run would execute this test routine. Interactive routines must be selected individually to be executed. When an interactive routine is selected, the (p) Loop and/or (x) All selectors will become non-selectable automatically, as this would be an invalid combination of control functions. See the Control Section discussion below for further information on interaction between the Control selectors and the Test Status routine selectors.
INDEX	Gives the status of each test routine. The status types are the same as those in the Subsystem Menu. Refer to the Subsystem Menu discussion.
FAULTS	Gives the number of failures for the routine. The number is reset to zero when Run is selected. Exceeding 65534 failures causes the overflow status, 65535+, to be displayed.
Execution Mark	This one-character field operates similar to the Subsystem Menu except that the <i>execution mark</i> refers only to one routine (i.e. there are no lower-level menus) and therefore can only take on the values of " " (space) or "*".
ADDRESS	Normally gives the internal memory or I/O address to which test data is written or read. Most useful for memory-related tests.
EXPECT	Gives the data expected by the test as a result of test stimulus. See the appropriate test description for details on test stimulus.
ACTUAL	Gives the actual data received by the test as a result of test stimulus.
MIN	Gives the minimum acceptable limit for the Actual test result data.
MAX	Gives the maximum acceptable limit for the Actual test result data.

5.2.4.2 Control Section

The following sections discuss special considerations which apply only to the Routine Menu level.

5.2.4.2.1 Interaction With Test Status Selectors

On entry into the Routine menu, one or more of the routine selectors in the Test Status Section (specifically, those that are interactive in some way) or the (p) Loop and/or (x) All selectors in the

Control Section, may be non-selectable so that invalid test conditions can not be attempted. Therefore, it may be necessary to set the (p) Loop and/or (x) All selectors Off to allow interactive routines in the Routine menu to be selectable. Similarly, it may be necessary to select a different routine in the Test Status Section (i.e. one that is not interactive) to allow the (p) Loop and/or (x) All selectors in the Control Section to be selectable.

5.2.4.2.2 Special Combinations - Cycle & Cycle-Halt Modes

Two combinations of the Control Section test execution modes (i.e. (p) Loop, (t) Terse, (x) All, and (s) Stop on Err) significantly increase the repetition time for most routines. These two combinations, referred to as Cycle and Cycle-Halt modes for short, help identify intermittent faults quicker and are beneficial when troubleshooting with an oscilloscope (by providing faster test pattern stimulus). The following table shows the appropriate test execution mode set-ups for Cycle and Cycle-Halt modes.

TABLE 5-1
Cycle and Cycle-Halt Mode Set-ups

	<u>(p) Loop</u>	<u>(t) Terse</u>	<u>(x) All</u>	<u>(s) Stop on Err</u>
Cycle	On	On	Off	Off
Cycle-Halt	On	On	Off	On

As can be seen, the only difference between Cycle and Cycle-Halt is that (s) Stop on Err is On for Cycle-Halt. This causes test execution to terminate on the first failure.

Normally, when only (p) Loop is On, the Executive processor commands a subsystem (and itself) to execute the selected test a single time (after which the test results for that single test execution are displayed) before commanding the subsystem to execute the test again. In other words, looping is being done through the Executive processor. However, in Cycle and Cycle-Halt modes, the Executive processor commands the subsystem to loop *locally*, and for it to keep track of the number of loops it is executing and the number of faults that it has found, up until the time test execution is terminated (via a test failure in Cycle-Halt mode or the normal front panel termination mechanisms in Cycle mode). Only after test execution is terminated will the test results be displayed. The net effect of this *local* looping in the subsystem is the speed-up of the test looping or cycling.

Cycle and Cycle-Halt modes will not appreciably improve cycle-times for routines which are normally slow in nature. Also, routines requiring interaction with other processors in the instrument may not have their cycle-times improved noticeably either.

5.2.5 Help Menu

As stated previously, pressing the (?) Help selector (in the Control Section of a diagnostic menu) displays a list of Field Replaceable Units (FRU's) for the currently selected routine. If the routine had a failure, then the error index code for that routine is also displayed. The FRU's are normally listed as the most-to-least probable cause for errors in the routine. However, in some cases, FRU's are listed according to their ease-of-replaceability due to the mechanical configuration of the instrument. In other words, FRU's are listed in a suggested order for replacement as a way to minimize the time required to identify the faulty FRU.

Within the Help Menu, the capability exists to scan forward to the next routine (keypad label "r") or to the next failure (keypad label "f") by touching the appropriate label in the lower portion of the

screen (which behaves like a simple keypad). Scanning continues in a circular fashion through all available subsystems. If a test failure can not be found when scanning for the next failure, a warning appears in the Prompt/Warning area of the display and no scanning movement takes place.

Upon exiting this menu, the instrument returns to the same menu level (i.e. Block, Area, etc.) as it was when the help function was invoked. However, the menu itself may be a different menu if any scanning was performed.

5.2.5.1 Front Panel Buttons

Two front panel buttons are active during Help Menus. They provide the following functions:

Touch Panel On/Off	Disables/enables alternately the touch panel from responding to user touches.
Hardcopy	Produces a hardcopy of the screen on an instrument-compatible printer. Refer to the section on Hardcopy for details.

5.2.6 Prompt/Keypad Menus

Some test routines which are user interactive in nature (i.e. " " blank index field), and which may require user input in some form (decimal, hex, or integer), cause the instrument to display special prompt and keypad menus (see Figure 5-5 below).

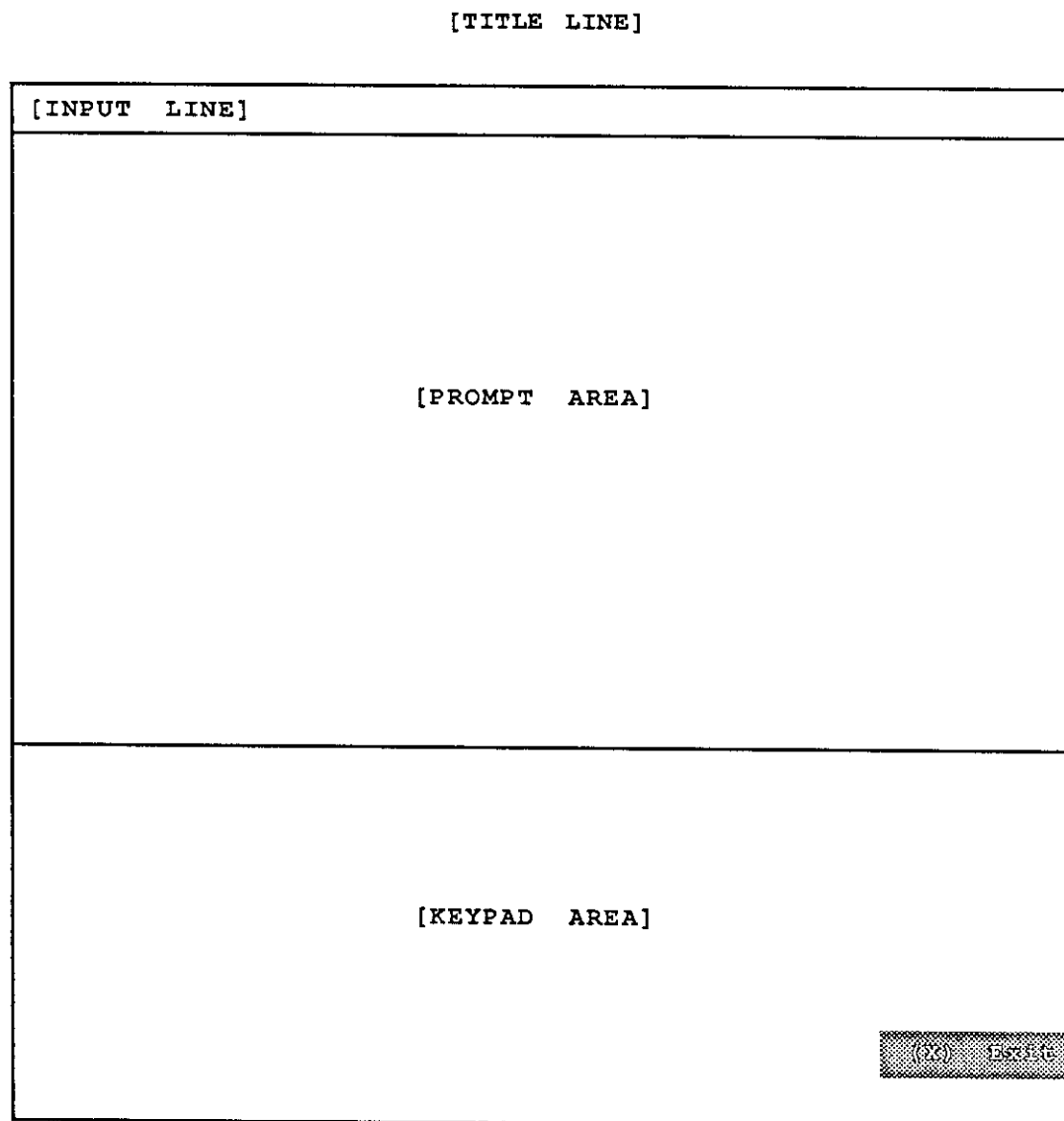


Figure 5-5. Prompt/Keypad Screen Layout

The areas of the prompt/keypad screen are used in the following way:

Title Line	This line is a logical title of the menu. It defaults to the summation of the subsystem, block, area, and routine names of the diagnostic menus. Operational errors/warnings appear on this line also.
Input Line	This area is where keypad (or terminal) input is displayed as it is entered.
Prompt Area	This area is where prompt lines are displayed. These text lines may prompt the user for input (in which case a keypad appears in the keypad area), tell the user to perform some operation, or show status of some executed action. At times, information being displayed in the prompt area causes the prompt area to scroll upward. The Acquisition Run/Stop button can be used to stop/start the scrolling action.
Keypad Area	This area is where most valid keypad (and terminal) selectors are shown, along with an exit selector. Depending on the type of user input being requested through the current prompt (and the stage of the entered value), some keypad selectors may be non-selectable.

5.2.6.1 Front Panel Buttons

Three front panel buttons are active during Prompt/Keypad Menus. They provide the following functions:

Touch Panel On/Off	Disables/enables alternately the touch panel from responding to user touches.
Hardcopy	Produces a hardcopy of the screen on an instrument-compatible printer. Refer to the section on Hardcopy for details.
Acquisition Run/Stop	Stops/starts information being displayed (and possibly scrolling) in the keypad prompt area.

5.2.7 Debugger Menu

The debugger function located in the Control Section of the Extended Diagnostics menus, when selectable, provides a simple hardware debugging editor (i.e. menu) that allows unrestricted low-level instrument hardware access. To make the debugger function selectable, or vice-versa, the Waveform button must be pressed five times followed by the Touch Panel On/Off button, all in succession. The Touch Panel On/Off button must then be pressed again to enable the front panel. The debugger function is normally not selectable in order to safeguard users from inadvertently corrupting crucial software parameters in RAM or NVRAM.

WARNING

Because there is unrestricted and unguarded access to all hardware, usage of the hardware debugger is potentially dangerous and should only be used when the instrument processor architectures and their associated memory and I/O layouts are known.

The hardware debugging editor (refer to Figure 5-6 below) allows the creation of a sequence or list of actions, each performing one of four basic operations (read, write, write/read, write/read/verify). Each operation has associated with it several parameters which further determine its behavior upon execution (via (T) Test or (L) Loop in the debugger menu). This part of the hardware debugging editor (upper left half of figure) is called the *Action Editor*. The *Action Editor* allows the valid entry and editing of all parameters associated with any individual action. The action that is shown in the *Action Editor* is one action of possibly many in the action list that is currently being edited.

The execution, creation, viewing, storage, and modification of action sequences or lists is done through the *List Editor* (upper right half of figure).

Executive Low-Level Hardware Debugger

7fc00									
(F) Action					(I) Insert		(A) Add		
2					Action		Action		
(O) Operation		(S) Address			(D) Delete		(C) Clear		
Write/Read		3fc00 hex			Action		Actions		
(R) Show Read		(L) Length			(P) Previous		(N) Next		
Yes		1 hex			Action		Action		
(W) Show Write		(B) Bits			(R) Recall		(S) Store		
No		0			List		List		
(M) Memory/IO		(V) Value			(M) Remove		(U) Undo		
Memory		00 hex			List		Lists		
(B) 8/16 Bits		(T) Times			(T) Test		(L) Loop		
8		1			Once		Stopped		
7	8	9	a	b	c	d	e	f	
4	5	6							
1	2	3	(<) Set to Min		Backspace				
		0							
0			(>) Set to Max		(CR) Enter		(X) Exit		
fffff									

Figure 5-6. Debugger Menu Example

The following sections will discuss each possible selector of the hardware debugging editor, starting with those in the *Action Editor* and finishing with the *List Editor* selectors.

5.2.7.1 Action Editor

The following selectors in the Debugger Menu comprise the *Action Editor* as shown in the figure above:

- | | |
|----------------|---|
| (#) Action | Allows random access to any action in the current list. When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the selection of the desired action. The status field shows which action is currently being displayed. |
| (o) Operation | Allows the selection of one of four basic operations: read, write, write/read, and write/read/verify. Each invocation of this selector causes the status to cycle through these four basic operations. This selector does not become highlighted when selected. |
| (r) Show Read | Toggles between Yes and No. This governs whether or not read operations are displayed when the current action list is executed (via (T) Test or (L) Loop). This selector is only selectable in read, write/read, and write/read/verify operations. This selector does not become highlighted when selected. |
| (w) Show Write | Toggles between Yes and No. This governs whether or not write operations are displayed when the current action list is executed (via (T) Test or (L) Loop). This selector is only selectable in write, write/read, and write/read/verify operations. This selector does not become highlighted when selected. |
| (m) Memory/IO | Toggles between Memory and IO. This governs whether the operation will be a memory or I/O port operation. If the selected subsystems' processor only allows memory operations, then the selector will be non-selectable and the status will default to Memory. If the selected subsystems' processor employs memory bank switching and the (m) Memory/IO selector is toggled to IO, then the (k) Bank selector will become non-selectable. The (m) Memory/IO selector does not become highlighted when selected. |
| (x) 8/16 Bits | Toggles between 8 and 16. This governs whether the operation will be done in bytes (8 bits) or words (16 bits). If the selected subsystems' processor only allows 8 bit operations, then this selector will be non-selectable and the status will default to 8. This selector does not become highlighted when selected. If the mode is switched from 8 to 16 bits, the (l) Length parameter may automatically be reduced so that the action does not exceed the memory or I/O space of the currently selected subsystems' processor. |
| (s) Address | Allows entry of the starting address. When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the address to be entered. The status field shows the starting address in hexadecimal. If the address is increased, the (l) Length parameter may automatically be reduced so that the action does not exceed the memory or I/O space of the currently selected subsystems' processor. |

- | | |
|------------|--|
| (l) Length | Allows entry of the length. When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the length to be entered. The status field shows the length in hexadecimal. |
| (k) Bank | Allows entry of the memory bank. When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the memory bank to be entered. The status field shows the memory bank in decimal. If the selected subsystems' processor does not employ memory bank switching, then this selector will be non-selectable. If it does employ bank switching and the (m) Memory/IO selector is set to IO, then the selector will also be non-selectable. |
| (v) Value | Allows entry of a hexadecimal value for operations which do writes (i.e. write, write/read, write/read/verify). When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the value to be entered. The status field shows the value in hexadecimal as either 8 or 16 bits, dependent on the (x) 8/16 Bits setting. If the (o) Operation selector is set to Read, then the (v) Value selector is non-selectable. |
| (t) Times | Allows entry of the number of times (iterations) the action should be executed before execution moves to the next action in the list. When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the number of times to be entered. The status field shows the number of times in decimal. |

5.2.7.2 List Editor

The following selectors in the Debugger Menu comprise the *List Editor* as shown in the figure above:

- | | |
|------------|---|
| (I) Insert | Causes another action item to be inserted into the list before the currently displayed action. The new actions items' parameters default to the previously displayed action item. This selector becomes non-selectable when the total number of actions in the current list reaches 99 or when the number of actions in the current list plus all of those actions which have been previously stored reaches or exceeds 350. This selector does not become highlighted when selected. |
| (A) Add | Causes another action item to be added to the list after the currently displayed action. The new action items' parameters default to the previously displayed action item. This selector becomes non-selectable when the total number of actions in the current list reaches 99 or when the number of actions in the current list plus all of those actions which have been previously stored reaches or exceeds 350. This selector does not become highlighted when selected. |
| (D) Delete | Causes the current action to be deleted. This selector is non-selectable when there is only one action in the current list. This selector does not become highlighted when selected. |
| (C) Clear | Causes all but the first action in the current action list to be deleted. This selector is non-selectable when there is only one action in the current list. This selector does not become highlighted when selected. |

- (P) Previous Causes the previous action in the current action list to be displayed in the *Action Editor*. This selector is non-selectable if the first action item is currently being displayed. This selector does not become highlighted when selected.
- (N) Next Causes the next action in the current action list to be displayed in the *Action Editor*. This selector is non-selectable if the last action item is currently being displayed. This selector does not become highlighted when selected.
- (S) Store Allows the current action list to be stored into the stored action list number that is displayed in the (S) Store status field. This selector becomes non-selectable when the total number of actions in the current list plus all of those actions which have been previously stored reaches or exceeds 350. This selector does not become highlighted when selected.

Recall of these stored action lists in the hardware debugger only remains intact while the user remains in Extended Diagnostics. If the instrument is powered down or normal operation is entered, all stored action lists will be lost and not available on the next entry into Extended Diagnostics. However, the hardware debugger menu may be exit'ed and re-entered after performing any other diagnostic operations or testing and the stored lists will be intact.

Stored action lists are stored (and recalled) according to the following subsystems' processor capabilities:

- 8/16 Bit Operations
- Memory Bank Switching
- Memory Space
- Input/Output (I/O) Space

What this means is that it is possible to have stored an action list from one place in the diagnostic menus (i.e. subsystem, block, etc.) and have it available for recall from another place in the diagnostic menus, given that the processors associated with the diagnostic menu selections (subsystem and block define the physical processor) have the same hardware debugging capabilities as shown above. This is the case, for instance, when storing an action list when one of the Mainframe Acquisition Subsystems is selected in the diagnostic menus and then selecting a Multi-Channel Unit Acquisition Subsystem from the diagnostic menus. When the hardware debugger is entered, the (R) Recall selector will be selectable because the action list stored previously is applicable to the current subsystems' processor configuration.

In addition, even though the hardware debugger might be exit'ed without storing the current action list, if, on the next entry to the debugger, the subsystems' processor configuration is the same as the last time it was exit'ed, the current action list will still be intact.

- (R) Recall Allows a previously stored action list to be recalled, after which it becomes the current action list. When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the stored action list number to be entered. This selector is non-selectable if there are no stored action lists which are appropriate for the current subsystems' processor.

- (M) Remove Allows a previously stored action list to be removed. When selected, it becomes highlighted and brings up a keypad in the lower keypad section of the screen which allows the stored action list number to be entered. This selector is non-selectable if there are no stored action lists which are appropriate for the current subsystems' processor. The status field (i.e. next stored action list number) of the (S) Store selector is reduced by one.
- (U) Null Allows all previously stored action lists which are appropriate for the current subsystems' processor to be removed. This selector is non-selectable if there are no stored action lists which are appropriate for the current subsystems' processor. This selector does not become highlighted when selected. The status field (i.e. next stored action list number) of the (S) Store selector is set by one.
- (T) Test Causes the hardware debugger to execute the current action list once. This selector does not become highlighted when selected. When execution occurs, the upper half of the screen (the *Action* and *List Editors*) is cleared and display output for each action, if enabled, is shown as they are executed. The display output, if enabled via the (r) Show Read and (w) Show Write selectors, for each action consists of the following items, in the order that they appear on each line from left-to-right:
- Action Number (from 1 to 99)
 - Read ('R') or Write ('W') Operation (one character)
 - Memory ('M') or I/O ('I') Operation (one character)
 - Memory Bank (one character if applicable)
 - Eight bytes or words (2 or 4 characters each)
 - Eight ASCII character-equivalents if 8 bit mode
- If there are more items than the prompt area of the screen will hold, the screen scrolls upward until the completion of execution, at which time scrolling stops and the debugger waits for an (cr) Enter or (X) Exit selection. If (cr) Enter is selected, the debugger returns to the hardware debugger editor. The (X) Exit selector exits back to the main diagnostic menus.
- Execution can be suspended by pressing the Acquisition Run/Stop button (or terminal keystroke 'Q'), at which time a prompt and keypad appears to see if the user wishes to continue or return to the hardware debugger editor. Hardcopy is available at this point, but not during the actual execution of actions.
- (L) Loop Causes the hardware debugger to execute the current action list continuously. This selector does not become highlighted when selected. Execution behavior is identical to (T) Test. To return to the hardware debugger editor, the Acquisition Run/Stop button (or terminal keystroke 'Q') must be pressed to suspend execution and bring up the continuation prompt.

5.2.7.3 Front Panel Buttons

Three front panel buttons are active during the hardware debugger. They provide the following functions:

Touch Panel On/Off	Disables/enables alternately the touch panel from responding to user touches.
Hardcopy	Produces a hardcopy of the screen on an instrument-compatible printer, except during debugger execution. Refer to the section on Hardcopy for details.
Acquisition Run/Stop	Suspends debugger execution as discussed under the (T) Test selector above.

5.2.8 Other Displays

Some test routines, when (r) **Run** is invoked, display information or graphics on the screen which do not conform to any of the aforementioned menu formats. Examples of these type of routines would be the front panel verification tests (see the routines in the Verify area of the Front Panel block of the Executive subsystem). The screen format and content for routines of this type are test specific and are covered only under their individual test description later in this manual.

5.3 Audible Indication of Test Termination

If (s) **Stop on Err** and (p) **Loop** are both On and a failure occurs (thereby causing test execution to halt), the speaker will produce two high-low beeps (similar to the Self-Test indication). This allows unattended operation while waiting for intermittent failures to occur. This feature applies to all four diagnostic menu levels (i.e. Subsystem, Block, Area, and Routine).

5.4 Front Panel Buttons, Knobs, and LEDs

Three front panel buttons and both knobs are active during the four main diagnostic menu levels (i.e. subsystem, block, area, and routine). They provide the following functions:

Touch Panel On/Off	Disables/enables alternately the touch panel from responding to user touches.
Hardcopy	Produces a hardcopy of the screen on an instrument-compatible printer. Refer to the section on Hardcopy for details.
Waveform	When used in the proper sequence with the Touch Panel On/Off button, enables the (D) Debugger selector. See the Debugger Section for the proper sequence of button presses.
Knobs	Controls the screen intensity.

NOTE

Within the four main diagnostic menu levels, all front panel buttons, regardless of whether they have functions associated with them or not, will cause diagnostic test execution to be terminated if they are pressed.

Various buttons are also active during other diagnostic menu displays, i.e. Help, Prompt/Keypad, and Debugger menus. The following table provides an overview of which buttons are active during all menus, including those described immediately above. Refer to the appropriate diagnostic menu section previously described for further information on the exact function of each button in each specific menu.

TABLE 5-2
Summary of Menus vs. Active Buttons & Knobs

<u>Menu</u>	<u>Active Buttons</u>
Subsys	Touch Panel On/Off, Hardcopy, Waveform, Knobs
Block	Touch Panel On/Off, Hardcopy, Waveform, Knobs
Area	Touch Panel On/Off, Hardcopy, Waveform, Knobs
Routine	Touch Panel On/Off, Hardcopy, Waveform, Knobs
Help	Touch Panel On/Off, Hardcopy
Prompt/Keypad	Touch Panel On/Off, Hardcopy, Acquisition Run/Stop
Debugger	Touch Panel On/Off, Hardcopy, Acquisition Run/Stop

While in the Extended Diagnostics mode, the Utility LED is on, the Front Panel On/Off LED is on or off (depending on whether the front panel is enabled or not), and the Acquisition Run/Stop LED is on. All other LEDs are off. These conditions may change during front panel test and verification routines.

5.5 Hardcopy Operation

The Hardcopy function of the instrument is available in a simplified form during Extended Diagnostics. This includes the four main diagnostic levels plus the Help, Prompt/Keypad, and Debugger menus. All menu items are depicted on the hardcopy. Menu items that are selected are printed in boldface. An Epson compatible printer must be connected to the rear-panel PRINTER port with a standard Centronics parallel interface cable. The compatibility requirements for hardcopy during diagnostics is less stringent than for normal operation due to the fact that diagnostic hardcopy currently does not support or use graphics capabilities of the attached printer. For printer connection and configuration information, refer to the 11801 or 11802 User Reference Manual.

To make a hardcopy of the screen display, press the front panel **HARDCOPY** button. The hardcopy of the displayed menu will include the following additional information at the top:

1. Instrument type; either **11801 DIGITAL SAMPLING OSCILLOSCOPE** or **11802 DIGITAL SAMPLING OSCILLOSCOPE**;
2. Current date and time; and
3. Unit identification of the mainframe.

Pressing the **HARDCOPY** button when no printer is connected or when the printer cannot print (e.g., off line, no paper, etc.) causes the following warning to be displayed:

Hardcopy absent or off line.

5.6 Remote Diagnostics

The instrument diagnostics can be remotely controlled with either of two modes. One is the front panel emulation mode, which provides near 100% control of the Extended Diagnostics from a terminal connected by an RS-232-C cable. The second is the system (normal operation) mode which allows a GPIB or an RS-232-C controller to invoke Self-Tests and Extended Diagnostics and to query the instrument for test status. The two modes are quite different so they are covered separately.

5.6.1 Front Panel Emulation Mode (RS-232-C Only)

The front panel emulation mode provides control of the Extended Diagnostics in a manner similar to front panel operation, except that terminal keystrokes are used instead of screen touches to select items. Only interactive tests, such as the Front Panel Verify tests, cannot be effectively performed from the remote terminal. This mode is useful when the display or touch panel is faulty or when the operator wishes to control the Extended Diagnostics remotely with a modem and phone link.

5.6.1.1 Entry

To use the front panel emulation mode, connect an ANSI 3.64 compatible terminal or Tektronix 410X/420X terminal to the instrument rear-panel RS-232-C port and put the 11801 or 11802 into the Extended Diagnostics mode. Make sure the ANSI 3.64 compatible terminal is in its ANSI compatible mode. Then, press one of the following keys on the terminal keyboard to transfer the instrument display to the terminal:

<u>Terminal</u>	<u>Entry Keystroke</u>
ANSI 3.64 compatible	"T" (shift T)
Tektronix 4105/4205	"K" (shift K)
Tektronix 4107/4109/4207/4208	"L" (shift L)

If the display is defective, the instrument can still be put into this mode of remote operation by having a Self-Test failure at power-up. The failure can be either an existing fault or an induced one (i.e., by pressing a button throughout the power-up cycle). When the Self-Tests finish with a failure, indicated by a double high-low beep, press "T" on the keyboard.

Upon transfer of control, the Subsystem menu is displayed on the terminal with all the information normally found on the front panel display. Each selectable menu item has an associated keystroke character in parentheses in front of it. The keystroke characters for the test status selections are reassigned when different menus are displayed, while the keystroke characters for the test control selections are the same for all menus.

In this mode, the touch panel is still active and mapped to selections on the terminal screen menu. Hence, touching the screen may cause a change in the remote terminal menu selections.

NOTE

Even while operating the extended diagnostics from the instrument touch panel and display (as normally would be the case), keystrokes from an attached terminal will cause the instrument to take appropriate key-equivalent actions.

5.6.1.2 Exit

To exit the front panel emulation mode, enter the "E" (shift E) character twice in succession (just as you would do on the touch screen). This causes the instrument to leave Extended Diagnostics and resume normal operation. Alternately, the "T", "K", or "L" characters, as described below, will transfer control of Extended Diagnostics back to the instrument front panel.

5.6.1.3 Non-Displayed Commands

The following keystrokes, available from a terminal attached to the RS-232-C port, do not appear in the menus on the instrument or terminal display. Their functions are as follows:

- | | |
|----------|--|
| B | <baud rate><CR> Changes the baud rate to one of the allowable baud rates (i.e., 110, 300 ..., 38400) from the default of 9600 (set with the Diagnostic Options Jumpers on the Input/Output Board). A carriage return must be entered after the new baud rate is entered in order to complete the entry. Note that the terminal baud rate should be changed after changing the instrument baud rate. |
| T | Toggles the screen output back and forth between the current screen display and an ANSI 3.64 compatible terminal. The terminal should initially be set manually to ANSI mode before transferring the display to the terminal the first time. |
| K | Toggles the screen output back and forth between the current screen display and a Tektronix 4105 or 4205 terminal. This provides a more exacting representation of the instrument display. A compatible color copier can be attached to the terminal to obtain a more representative hardcopy of the diagnostic display (vs. the normal hardcopy button). |
| L | Toggles the screen output back and forth between the current screen display and a Tektronix 4107, 4109, 4207, or 4208 terminal. This provides the optimum representation of the instrument display on a terminal screen. A compatible color copier can be attached to the terminal to obtain a more representative hardcopy of the diagnostic display (vs. the normal hardcopy button). |
| H | Produces a hardcopy of the current diagnostic display, as detailed earlier in the Hardcopy section. |
| O | Disables/enables the touch panel from responding to user touches. This keystroke is equivalent to the Touch Panel On/Off button. |
| W | When used in the main diagnostic menus and in the proper sequence with the Touch Panel On/Off button, enables the (D) Debugger selector. See the Debugger Section for the proper sequence of button presses. This keystroke is equivalent to the Waveform button. |

- Q** When used in interactive prompt/keypad menus, stops/starts information being displayed (and possibly scrolling) in the keypad prompt area. When used in the hardware debugger editor, suspends debugger execution and brings up a continuation prompt so the user may return to the hardware debugger editor. This keystroke is equivalent to the Acquisition Run/Stop button in these menus.

5.6.2 System Mode (GPIB and RS-232-C)

The system (i.e. normal) operating mode allows either the Self-Tests or Extended Diagnostics to be invoked with external interface commands. The commands are as follows:

<u>Command</u>	<u>Argument</u>	<u>Notes</u>
TEST	[XTND MAN]	Set only
DIAG		Query only

For detailed external interface, syntax, and usage information, refer to the 11801 or 11802 Programmer's Reference Manual.

5.6.2.1 TEST & TEST XTND Commands

TEST without arguments initiates the Self-Tests. TEST with the argument XTND initiates the Extended Diagnostics.

When the 11801/11802 receives a TEST command, the instrument goes through a pseudo power-down sequence (i.e., saves current settings), performs the testing requested, then does a power-up sequence, regardless of whether or not faults were found. In the system mode, a self-test failure does not cause the instrument to enter the Extended Diagnostics mode.

When requested by the TEST XTND command, Extended Diagnostics are automatically configured to run all automatic tests, and then to exit back to normal operation upon completion of testing. The touch panel is disabled during Extended Diagnostics, which is indicated by the Touch Panel On/Off LED being off and by the following message in the Prompt/Warning Section of the display:

Front panel locked out.

Completion of diagnostic testing is signaled by one of the following event codes:

<u>Event Code</u>	<u>Explanation</u>
460	Self- or extended-test diagnostics were completed successfully.
394	Self- or extended-test diagnostics were completed and failed.

If RQS is ON when the TEST command is executed, an SRQ interrupt will signal test completion to the controller. Further error information is available with the DIAG query command.

TEST deletes all stored waveforms upon execution.

5.6.2.2 TEST MAN Command

TEST MAN initiates the Extended Diagnostics in the same way extended diagnostics are entered through the front panel Utility menu. When the 11801/11802 receives this command, the instrument goes through a pseudo power-down sequence (i.e., saves current settings), then enters the Extended Diagnostics menu system and waits for user input from the front panel or RS-232-C port.

Note

If the TEST MAN command is issued from the GPIB port with no attached terminal on the RS-232-C port, the only way for the operator to exit Extended Diagnostics is to manually touch the (E) Exit selector twice in succession. Therefore, it is suggested that this command only be used when the instrument is being driven from the RS-232-C port.

The TEST MAN command is not documented in the 11801 or 11802 Programmer Reference Manuals due to the fact that it does not conform to Tektronix Codes & Formats guidelines. It is a special diagnostic feature whose primary use is to provide the remote diagnostic operator (i.e. troubleshooter) a convenient way of accessing the extended diagnostics terminal mode through the normal RS-232-C mode of operation without requiring remote assistance (i.e. the need to have someone at the instrument site manually switch the instrument from normal operation into Extended Diagnostics).

After issuing the TEST MAN command, the remote operator may switch the instrument display to the terminal display by issuing one of the transfer display commands described in the Remote Diagnostics, Non-Displayed Commands section.

5.6.2.3 DIAG? Query Command

The DIAG? query returns pass/fail information from the most recent invocation of Self-Tests or Extended Diagnostics. Power-up failure information can also be obtained with this query.

Failure information includes error index codes. A query after Self-Tests have run will return only one error index code per subsystem. A query after Extended Diagnostics have run, however, will return up to 206 error index codes.

Pass/fail status will be in different forms depending on instrument configuration and whether or not the power-up Self-Tests were bypassed via the Diagnostics Options Jumpers.

5.6.2.3.1 Diagnostics Passed Response

When the internal diagnostics finds no faults, the DIAG? response reflects the instrument configuration.. For example, if no faults were found and there are no SM-11 Multi-Channel Units attached (11801 only), the query response will be

DIAG PASSED:"a????,b????,c????,d????" (11801 Only)

DIAG PASSED:"NONE" (11802 Only)

If no problems were found and all possible SM-11's are attached (11801 only), the query response will be

DIAG PASSED:"NONE" (11801 & 11802)

5.6.2.3.2 Diagnostics Failed Response

When the internal diagnostics finds one or more faults, the DIAG? response will be similar to the following example:

DIAG FAILED:"E1311,c????,E1711,E1721,E1731" (11801 only)

DIAG FAILED:"E1311,E1711,E1721,E1731" (11802)

The first and third through the fifth entries (in the 11801 example) are error index codes from the Executive subsystem. The second entry indicates that there were no acquisition systems resident in Multi-Channel Unit C (e.g. there was no unit C attached) or that none of the acquisition systems in Multi-Channel Unit C (if attached) communicated with the Timebase system on power-up.

5.6.2.3.3 Diagnostics Bypassed Response

When the instrument is powered up with the Self-Tests bypassed with the Diagnostics Options Jumpers the query response will be:

DIAG BYPASSED

This query response will not occur when the Self-Tests are initiated during normal operation since Self-Tests can not be bypassed at that time.

6 Test Descriptions

6.1 Kernel Page Layout

Kernel test descriptions are laid out to provide easy test identification and comprehensive test descriptions that are easy to follow. The page elements, in their order down the page, are as follows:

Page Header	Test identification is simplified by the subsystem name and routine name at the top of each page. The format is as follows: <div style="display: flex; justify-content: space-between;"><div>Subsystem Name</div><div>Routine Name (Kernel Error Index)</div></div> This information makes finding the test description for a given kernel error index code easier because the kernel error index codes, located at the outside edge of each page, are easy to see while thumbing through the test pages. Also, it lets you know clearly what test is being discussed on each page of a long test description.
Routine	This name is the same as that used in the page header and provides a general idea of the circuit being tested.
Overview	Provides a brief description of how the circuit was setup and exercised and it refers to schematic diagrams covering the exercised circuits.
Operator Procedure	This label is included only for interactive tests. When included in a test description, it provides setup and procedural instructions to verify, calibrate, or troubleshoot a specific circuit.
Description	Provides a detailed, step-by-step description of how the test routine exercises the circuit, including information about the operation of important signal lines, components, and circuits.
Error Index	This label is followed by a two digit error index code. The associated discussion describes the possible failure modes for the circuit tested.
See Also	Provides other references when necessary or helpful. Some examples are references to Checks and Adjustments procedures in the Service Reference Manual or to interactive diagnostic tests used to verify apparent failures.
Caveats	This entry gives other conditions that could cause the test to fail, such as improper instrument setup for the test routine.

6.2 Self-Test/Extended Page Layout

Self-Test/Extended test descriptions are laid out to provide easy test identification and comprehensive test descriptions that are easy to follow. The page elements, in their order down the page, are similar to those described in the Kernel Page Layout section above, with the following differences:

Page Header	Test identification is simplified by the Block, Area, and Routine information at the top of each page. The format is as follows:
	<div>Block Name</div> <div>Area Name</div> <div>Routine Name (Error Index)</div>
Routine	This name is the same as that used in the Extended Diagnostics routine menus.
Error Index	This label is followed by a five digit error index code. The associated discussion describes the failure mode for the circuit tested.
Page Footer	The subsystem to which the test belongs is shown at the bottom of the page.

6.3 Notation Used

In the test descriptions, certain notation conventions are used to represent signal line names, component names, and circuit names. Circuit names, which coincide with names of circuit areas on schematics, are shown in bold type, as in

TBC Interrupt Latch <14>

Signal lines and component names appear as upper case, as in

INTR(H) U210A-5 <14> or U210A-1(L):L <14>

Signal lines and component names also have some other features in common. A signal's "true", "high", "active", or "on" state is in parentheses after its name or number. The dash and number following the "U" number specify a particular pin number of the component. The numbers at the end of signal names appearing in "<>" specify the schematic diagram on which the component can be found. Finally, on the right example above, the ":L" indicates the signal is being driven low. It is used as in "Set U210A-1(L):L <14>"; which says that pin 1 of U210A is driven low, its true state.

6.4 Executive Kernel Overview

The following list gives an overview of the kernel tests that the Executive processor executes at power-up before Self-Tests run. The kernel tests are shown in the order that they are executed and documented later in this manual. This overview can be used to help locate the desired test description.

1F	DRAM Data Lines
1E	DRAM Byte Access
1D	DRAM Address/Data
1C	Bank Select & Memory Configuration
1B	ROM U240 Location
1A	ROM U240 Checksum
19	ROM U250 Location
18	ROM U250 Checksum
17	RAM/ROM U230 Location
16	RAM/ROM U230 Checksum
15	RAM/ROM U240 Location
14	RAM/ROM U240 Checksum
13	Timer 0 Interrupt
12	Timer 1 Interrupt
11	Timer 2 Interrupt
10	80287 Math Coprocessor
0F	DMA Controller Interrupt
0E	Front Panel Controller Interrupt
0D	Serial Data Interface Interrupt
0C	Real Time Clock Interrupt
0B	GPIB Controller Interrupt
0A	Printer Controller Interrupt
09	Std RS232 Controller Interrupt
08	MMU Display Talk Request Interrupt
07	MMU SAG Interrupt
06	MMU RAG Interrupt
05	DMA 0 Transfer
04	Std RS232 External Loopback (forced by I/O Bd straps only)

6.5 Display Kernel Overview

The following list gives an overview of the kernel tests that the Display processor executes at power-up before Self-Tests run. The kernel tests are shown in the order that they are executed and documented later in this manual. This overview can be used to help locate the desired test description.

FF	RAM Data Lines
FE	RAM Address/Data
FD	ROM U612 Location
FC	ROM U602 Location
FB	ROM U612 Checksum
FA	ROM U602 Checksum
F9	DMA 0
F8	DMA 1
F7	Timer 0
F6	Timer 1
F5	Timer 2
F4	Executive Communication

6.6 Timebase Kernel Overview

The following list gives an overview of the kernel tests that the Timebase processor executes at power-up before Self-Tests run. The kernel tests are shown in the order that they are executed and documented later in this manual. This overview can be used to help locate the desired test description.

1	CPU Registers
2	ROM U400 Location/Checksum
3	ROM U410 Location/Checksum
4	Static RAM Data Lines
5	Static RAM Address/Data
6	Processor Peripherals
7	MMU Control
8	MMU Handshake
9	MMU Messages
A	MMU Addressing
B	Acquisition Configuration
C	Executive Communication

6.7 Acquisition Kernel Overview

The following list gives an overview of the kernel tests that the Acquisition processors, both those in the mainframe and those existing in any attached SM-11 Multi-Channel Units, execute at power-up before Self-Tests run. The kernel tests are shown in the order that they are executed and documented later in this manual. This overview can be used to help locate the desired test description.

1	EPROM U611 Location/Checksum
2	RAM Data/Address
3	Timebase Communication

6.8 Executive Self-Test/ Extended Overview

The following list gives an overview of the Executive subsystem Extended Diagnostic menus and tests. They are shown in the exact order and manner that they appear in the Extended Diagnostic menus and are documented later in this manual. This overview can be used to help locate the desired test description.

Memory (E1XXX)

ROM Loc Lo (E11XX)

- U250 (E111X)
- U630 (E112X)
- U600 (E113X)
- U612 (E114X)
- U620 (E115X)
- U210 Mem Xpn (E116X)
- U240 Mem Xpn (E117X)

ROM Loc Hi (E12XX)

- U240 (E121X)
- U730 (E122X)
- U700 (E123X)
- U712 (E124X)
- U720 (E125X)
- U220 Mem Xpn (E126X)
- U230 Mem Xpn (E127X)

ROM Cksum Lo (E13XX)

- U250 (E131X)
- U630 (E132X)
- U600 (E133X)
- U612 (E134X)
- U620 (E135X)
- U210 Mem Xpn (E136X)
- U240 Mem Xpn (E137X)

ROM Cksum Hi (E14XX)

- U240 (E141X)
- U730 (E142X)
- U700 (E143X)
- U712 (E144X)
- U720 (E145X)
- U220 Mem Xpn (E146X)
- U230 Mem Xpn (E147X)

RAM Refresh (E15XX)

- Rate (E151X)

Dynamic RAM (E16XX)
 Config (E161X)
 Data Lines (E162X)
 Address/Data (E163X)

NVRAM NoBank (E17XX)
 Battery (E171X)
 Data Lines (E172X)
 Address/Data (E173X)

NVRAM Banks (E18XX)
 Battery (E181X)
 Data Lines 0 (E182X)
 Addrs/Data 0 (E183X)
 Data Lines 2 (E184X)
 Addrs/Data 2 (E185X)

Control (E2XXX)

Intrrpt Ctrl (E21XX)
 Master (E211X)
 Slave 1 (E212X)
 Slave 3 (E213X)

Timers (E22XX)
 Timer 0 (E221X)
 Timer 1 (E222X)
 Timer 2 (E223X)
 Diagn Signal (E224X)

TimerIntrpts (E23XX)
 Timer 0 (E231X)
 Timer 1 (E232X)
 Timer 2 (E233X)

MPU Waits (E24XX)
 Zero Wait (E241X)
 One Wait (E242X)
 Two Wait (E243X)
 Four Wait (E244X)

ROM Waits (E25XX)
 Zero Wait (E251X)

Math Coproc (E26XX)
 Floating Pt (E261X)

DMA's (E27XX)
 DMA 0 (E271X)
 Interrupt (E272X)

Front Panel (E3XXX)

Control (E31XX)

- RAM (E311X)
- RAM Control (E312X)
- Interrupt (E313X)

Hard Keys (E32XX)

- Open (E321X)

Soft Keys (E33XX)

- Row Open (E331X)
- Column Open (E332X)
- Row Close (E333X)
- Column Close (E334X)

Knobs (E34XX)

- Upper Knob (E341X)
- Lower Knob (E342X)

Verify

- Hard Keys
- Soft Keys
- Knobs

Input/Output (E4XXX)

Temp Sensor (E41XX)

- Comparator (E411X)

RealTime Clk (E42XX)

- Counting (E421X)
- Interrupt (E422X)
- Calibrate

Tone Gen

- Ramp Tone

Printer (E44XX)

- Loopback (E441X)
- Interrupt (E442X)
- Pattern

RS232 (E45XX)

- Loopback (E451X)
- Baud Rate (E452X)
- Error Gen (E453X)
- Interrupt (E454X)
- Extern Loop (E455X)

GPIB (E46XX)

- Intrpt Reset (E461X)
- Reset Status (E462X)
- Data Lines (E463X)
- Interrupt (E464X)

Subsys Comm (E5XXX)

MMU Control (E51XX)

- Status Reg (E511X)
- Arbitration (E512X)
- Refresh (E513X)
- Dtalk Intrpt (E514X)
- SAG Compare (E515X)
- SAG Adder (E516X)
- SAG Intrpt (E517X)
- RAG Regs (E518X)
- RAG Intrpt (E519X)

Waveform RAM (E52XX)

- Size (E521X)
- Data Lines (E522X)
- Address/Data (E523X)

WCA Control (E53XX)

- Reset (E531X)
- Allnull (E532X)
- M/Mund (E533X)
- Idle (E534X)
- Comprss Null (E535X)
- Xparent Null (E536X)
- Comprss Over (E537X)
- Xparent Over (E538X)
- Comprss Undr (E539X)
- Xparent Undr (E53AX)
- Non Special (E53BX)

WCA Cmprssor (E54XX)

- Max Special (E541X)
- Max DataLine (E542X)
- Min DataLine (E543X)
- Min Special (E544X)

WCA Adder (E55XX)

- Offset (E551X)
- Data Paths (E552X)
- Overrange (E553X)
- Underrange (E554X)

- MainFrm Comm (E56XX)
 - Display (E561X)
 - Time Base (E562X)
 - TBC Intrpt (E563X)

- SDI (E57XX)
 - Interrupt (E571X)

6.9 Display Self-Test/ Extended Overview

The following list gives an overview of the Display subsystem Extended Diagnostic menus and tests. They are shown in the exact order and manner that they appear in the Extended Diagnostic menus and are documented later in this manual. This overview can be used to help locate the desired test description.

Dsy Control (D1XXX)

- ROM Location (D11XX)
 - U612 (D111X)
 - U602 (D112X)

- ROM Checksum (D12XX)
 - U612 (D121X)
 - U602 (D122X)

- Static RAM (D13XX)
 - Data Lines (D131X)
 - Address/Data (D132X)

- Timers (D14XX)
 - Timer 0 (D141X)
 - Timer 1 (D142X)
 - Timer 2 (D143X)

- DMA's (D15XX)
 - DMA 0 (D151X)
 - DMA 1 (D152X)

- Exec Intrfce (D16XX)
 - Command Port (D161X)
 - DMA Access (D162X)
 - Wavefrm Port (D163X)
 - Attributes (D164X)

Video Gen (D2XXX)

Timing (D21XX)

- Trace CAS (D211X)
- BSRLOAD (D212X)
- Bit1 CAS (D213X)
- ACCLK (D214X)
- Trace RAS (D215X)
- CRTC R/W (D216X)
- B2HE (D217X)
- B1LE (D218X)
- VOLE (D219X)
- Crastersync (D21AX)
- Cfieldsync (D21BX)
- Dispen (D21CX)

Address Mux (D22XX)

- MPU Address (D221X)
- CRTC Address (D222X)
- CRTC R/W (D223X)

Bit Plane 1 (D23XX)

- Data Lines (D231X)
- Address/Data (D212X)

Bit Plane 2 (D24XX)

- Data Lines (D241X)
- Address/Data (D242X)

Trace Plane (D25XX)

- Data Lines (D251X)
- Address/Data (D252X)

VRS Gen (D26XX)

- Single Axis (D261X)
- Dual Axis (D262X)
- Color Map (D263X)
- Video Shfter (D264X)
- Priority (D265X)

CRT Driver

- Stimulus

6.10 Timebase Self-Test/ Extended Overview

The following list gives an overview of the Timebase subsystem Extended Diagnostic menus and tests. They are shown in the exact order and manner that they appear in the Extended Diagnostic menus and are documented later in this manual. This overview can be used to help locate the desired test description.

Test Descriptions

TBC Control (T1XXX)

ROM Location (T11XX)

U300 (T111X)

U310 (T112X)

U400 (T113X)

U410 (T114X)

ROM Checksum (T12XX)

U300 (T121X)

U310 (T122X)

U400 (T123X)

U410 (T124X)

Static RAM (T13XX)

Data Lines (T131X)

Address/Data (T132X)

Battery (T133X)

Interrupts (T14XX)

Internal (T141X)

DMA's (T15XX)

DMA0 (T151X)

DMA1 (T152X)

Exec Intrfce (T16XX)

Control (T161X)

Handshake (T162X)

Messages (T163X)

Waveforms (T164X)

Addressing (T165X)

Timers (T17XX)

Timer 2 (T171X)

Timer 0 (T172X)

Timer 1 (T173X)

Strobe Gen (T2XXX)

VCO (T21XX)

Control (T211X)

Ramp DACS

Strbe Digitl (T22XX)

Control (T221X)

Pipeline (T222X)

Slow Count (T223X)

Corct Table (T224X)

Strbe Analog (T23XX)
 SDS (T231X)
 Fast Count (T232X)
 Holdoff (T233X)
 Ref Clocks (T234X)
 SDS Plot
 Ramp DACS

Trigger (T24XX)
 Control (T241X)
 Level (T242X)
 Dly Adj (T243X)
 Calibrator
 Ramp DACS

Exerciser
 Strobes
 Trigger
 VCO
 Int Clk
 BH Adjust
 Execute

M/F I/F (T3XXX)

Comm (T31XX)
 Acq 1 (T311X)
 Acq 2 (T312X) - refer to previous Comm Acq 1 test

Shared Mem (T32XX)
 Acq 1 (T321X)
 Acq 2 (T322X) - refer to previous Shared Mem Acq 1 test

Strobe Gate (T33XX)
 Acq 1 (T331X)
 Acq 2 (T332X) - refer to previous Strobe Gate Acq 1 test

Strobe Sense (T34XX)
 Acq 1 (T341X)
 Acq 2 (T342X) - refer to previous Strobe Sense Acq 1 test

Arbitor Test
 Acq 1
 Acq 2 - refer to previous Arbitor Test Acq 1 test

MCU A (T4XXX) - refer to M/F I/F Acq 1 tests (T3XXX) above for each appropriate routine

MCU B (T5XXX) - refer to M/F I/F Acq 1 tests (T3XXX) above for each appropriate routine

MCU C (T6XXX) - refer to M/F I/F Acq 1 tests (T3XXX) above for each appropriate routine

MCU D (T7XXX) - refer to M/F I/F Acq 1 tests (T3XXX) above for each appropriate routine

6.11 Mainframe & SM-11 Acquisition Self-Test/ Extended Overview

The following list gives an overview of the Acquisition subsystem Extended Diagnostic menus and tests for the Acquisitions present in the mainframe (Main Acq) and those existing in any attached SM-11 Multi-Channel Units (MCU A Acq, MCU B Acq, MCU C Acq, and MCU D Acq). They are shown in the exact order and manner that they appear in the Extended Diagnostic menus and are documented later in this manual. This overview can be used to help locate the desired test description.

Acq 1 (m1XXX)

Memory (m11XX)

- ROM Loc (m111X)
- ROM Check (m112X)
- RAM Data (m113X)
- RAM Address (m114X)

Interrupt (m12XX)

- Timer (m121X)

Control (m13XX)

- Dig Cntrl (m131X)

Signal Path (m14XX)

- Diag ADC (m141X)
- Strobes (m142X)
- Gain (m143X)
- Offset (m144X)
- Correction (m145X)
- Sig Path A (m146X)
- Sig Path B (m147X)

Measurement (m15XX)

- Ref Level (m151X)
- Filter A (m152X)
- Filter B (m153X)
- A->B B->A (m154X)

Sample Head (m16XX)

- Cntrl Func 1 (m161X)
- Cntrl Func 2 (m162X)
- Cntrl Func 3 (m163X)
- Cntrl Func 4 (m164X)
- Cntrl Func 5 (m165X)
- EEPROM (m166X)

Exercisers

- Calibration
- DACs
- Signal Path
- Meas Sys
- Registers
- Ms Lvl Ver

Acq 2 (m2XXX) - refer to Acq 1 tests (m1XXX) above for each appropriate routine

Acq 1-8 (a[1-8]XXX) - refer to Acq 1 tests (m1XXX) above for each appropriate routine

Acq 1-8 (b[1-8]XXX) - refer to Acq 1 tests (m1XXX) above for each appropriate routine

Acq 1-8 (c[1-8]XXX) - refer to Acq 1 tests (m1XXX) above for each appropriate routine

Acq 1-8 (d[1-8]XXX) - refer to Acq 1 tests (m1XXX) above for each appropriate routine

Routine Name	DRAM Data Lines
Overview	This test verifies the data lines to DRAM <30> by performing a "walking one's" test on DRAM address 00000hex.
Description	<ol style="list-style-type: none">1. Perform a "walking one's" test on DRAM address 00000hex.<ul style="list-style-type: none">• Write the pattern 8000hex to address 00000hex. Read the same address and verify that the pattern was 8000hex. Continue this write/read/verify sequence with patterns 4000hex, 2000hex, 1000hex, 0800hex, 0400hex, 0200hex, 0100hex, 0080hex, 0040hex, 0020hex, 0010hex, 0008hex, 0004hex, 0002hex, 0001hex and 0000hex.
Error Index 1F	The pattern read from DRAM <30> address 00000hex was not the pattern written.

Routine Name DRAM Byte Access

Overview This test verifies the DRAM R/W Control's <30> ability to address odd and even bytes of DRAM <30> memory locations separately by writing to locations 00001hex and 00002hex.

Description

1. Perform byte access test.
 - Write/read/verify pattern AAhex to DRAM address 00001hex.
 - Write/read/verify pattern 55hex to DRAM address 00002hex.
 - Write/read/verify pattern CChex to DRAM address 00001hex.

Error Index 1E The pattern read from one of the byte locations was not the same as the pattern written.

Routine Name DRAM Address/Data

Overview This test verifies the address lines and data integrity of DRAM <30> address range 00000-2FFFF_{hex} by performing a RAM test on this address range.

Description

1. Verify address range 00000-0FFFF_{hex} (64k bytes). Terminate test if any verify operation fails.
 - Fill address range 00000-0FFFF_{hex} with the pattern AAAA_{hex}.
 - Read and verify address 00000_{hex} for AAAA_{hex}. If so, write CCCC_{hex} to address 00000_{hex}. Increment address and continue this read/verify/write sequence until address 0FFFF_{hex} is reached.
 - Repeat the read/verify/write sequence, starting again at address 00000_{hex} for CCCC_{hex}, 5555_{hex}, and 0000_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern).
2. Repeat step 1 for address ranges 10000-1FFFF_{hex} and 20000-2FFFF_{hex}.
3. Verify address lines A16 and A17.
 - Write 0000_{hex} to address 00000_{hex}, 0001_{hex} to address 10000_{hex}, and 0002_{hex} to address 20000_{hex}.
 - Read and verify that address 20000_{hex} contains 0002_{hex}, 10000_{hex} contains 0001_{hex}, and 00000_{hex} contains 0000_{hex}.

Error Index 1D The pattern read from a DRAM address location was not the pattern written. On failure, the DRAM Data Lines test (1F) is repeatedly performed on the failed address location.

Routine Name Bank Select and Memory Configuration

Overview This test verifies **DRAM Configuration** <30> jumper selections and the software bank selectability through the **Bank Decode/Select** <27> circuit.

Description

1. Read the **Memory Configuration Readback** U832 <29>. Only the upper four bits of this port represent the jumper positions. The lower four bits represent the current bank selection.
2. Verify the position of the EPROM-size selection jumper J541 <29> for 27512 type EPROMs.
3. Verify the position of the virtual or real memory addressing jumper J520 <30> for real memory addressing.
4. Verify the combination of RAM device type and number of RAM banks jumpers J501 <30> and J521 <30>, respectively, for 64k x 4 RAM device type and two banks.
5. Store the current bank by reading **Bank Decode/Select** U530A <27>.
6. Verify bank selectability by doing a "walking one's" test on the 4-bit bank select register U520 <27>.
 - Write patterns 01hex, 02hex, 04hex, and 08hex to **Bank Decode/Select** U520 <27>. Read and verify the written patterns from **Bank Decode/Select** U530A <27> and **Memory Configuration Readback** U832 <29> (only the lower four bits).
7. Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index 1C The 27512 EPROM has not been selected through J541; or real mode selection has not been made through J520; or the RAM device and number of banks combination selected through J501 and J521, respectively, are not appropriate; or the bank select mechanism is not functioning properly.

Routine Name ROM U240 Location

Overview This test verifies that ROM U240 <28> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the EPROM bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Exclusive-or the contents of byte locations *E0009hex* and *E000Bhex* and verify that the result is *FFhex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the EPROM bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.

Error Index 1B The bytes at *E0009hex* and *E000Bhex* were not complementary or the location byte did not match the known value.

Routine Name ROM U240 Checksum

Overview This test verifies the integrity of ROM U240 <28> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations E0009_{hex} and E000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U240 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index 1A The bytes at E0009_{hex} and E000B_{hex} were not complementary or the computed checksum did not match the stored checksum.

Routine Name ROM U250 Location

Overview This test verifies that ROM U250 <28> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations E0008_{hex} and E000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index 19 The bytes at E0008_{hex} and E000A_{hex} were not complementary or the location byte did not match the known value.

Routine Name ROM U250 Checksum

Overview This test verifies the integrity of ROM U250 <28> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U250 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index 18 The bytes at E0008*hex* and E000A*hex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name RAM/ROM U230 Location

Overview This test verifies that **RAM/ROM U230 <32>** is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Exclusive-or the contents of byte locations **80009_{hex}** and **8000B_{hex}** and verify that the result is **FF_{hex}**.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.

Error Index 17 The bytes at **80009_{hex}** and **8000B_{hex}** were not complementary or the location byte did not match the known value.

Routine Name RAM/ROM U230 Checksum

Overview This test verifies the integrity of RAM/ROM U230 <32> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Exclusive-or the contents of byte locations *80009hex* and *8000Bhex* and verify that the result is *FFhex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Perform a checksum on all the bytes in U230 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.

Error Index 16 The bytes at *80009hex* and *8000Bhex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name RAM/ROM U240 Location

Overview This test verifies that **RAM/ROM U240 <32>** is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Exclusive-or the contents of byte locations *80008hex* and *8000Ahex* and verify that the result is *FFhex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.

Error Index 15 The bytes at *80008hex* and *8000Ahex* were not complementary or the location byte did not match the known value.

Routine Name RAM/ROM U240 Checksum

Overview This test verifies the integrity of RAM/ROM U240 <32> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the RAM/ROM <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations 80008_{hex} and 8000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the RAM/ROM <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U240 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index 14 The bytes at 80008_{hex} and 8000A_{hex} were not complementary or the computed checksum did not match the stored checksum.

Routine Name Timer 0 Interrupt

Overview This test verifies the **Timer U822 <25> COUNTER 0** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Clear the **Timer COUNTER 0** interrupt.
 - Write *00hex* to **Timer Configuration Logic U720 <25>**, *34hex* to **Timer U822 <25> Control Word Register**, and then toggle **Timer 0 Interrupt Reset U812B-13 <25>**.
2. Verify that there is no **Timer COUNTER 0** interrupt pending at the **Interrupt Controllers MASTER U350 <28>**.
 - Read **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that **IR2 U350-20 <28>** is low.
3. If there is no **Timer COUNTER 0** interrupt pending, then enable **Timer COUNTER 0** interrupt by setting bit 2 of the **Interrupt Controllers MASTER U350 <28> Mask Register** low.
4. Complete the programming for **Timer COUNTER 0** with *80hex* as the starting count.
 - Write *80hex* and *00hex* to the **Count Register low and high bytes**, respectively. This starts the counter.
5. Perform a software delay.
6. Verify that the **Timer COUNTER 0** interrupt did occur.
7. Disable the **Timer COUNTER 0** interrupt by setting bit 2 of the **Interrupt Controllers MASTER U350 <28> Mask Register** high.
8. Restore the **Timer COUNTER 0** for the normal operating firmware.

Error Index 13 The **Timer COUNTER 0** interrupt did not occur or it could not be cleared at the **Interrupt Controllers MASTER U350<28>**.

Routine Name Timer 1 Interrupt

Overview This test verifies the **Timer** U822 <25> COUNTER 1 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Configure the **Timer** COUNTER 1 to count 6 MHz clock.
 - Write *00hex* to **Timer Configuration Logic** U720 <25> so that G1 U822-14 <25> is high and PCLK(H) is connected to CLK1 U822-15 <25>.
2. Clear **Timer** COUNTER 1 interrupt.
 - Write *70hex* to the Control Word Register.
3. Verify that there is no **Timer** COUNTER 1 interrupt pending at the **Interrupt Controllers SLAVE 3** U370 <28>.
 - Read **Interrupt Controllers SLAVE 3** U370 <28> INTERRUPT REQUEST REGISTER and verify that IR0 U370-18 is low.
4. If there is no **Timer** COUNTER 1 interrupt pending, then enable the **Timer** COUNTER 1 interrupt by setting bit 0 of **Interrupt Controllers SLAVE 3** U370 <28> Mask Register and bit 3 of **Interrupt Controllers MASTER** U350 <28> Mask Register low.
5. Complete programming the **Timer** COUNTER 1 with *80hex* as the starting count.
 - Write *80hex* and *00hex* to the Count Register low and high bytes, respectively. This starts the counter.
6. Perform a software delay.
7. Verify that **Timer** COUNTER 1 interrupt did occur.
8. Disable the **Timer** COUNTER 1 interrupt by setting bit 0 of **Interrupt Controllers SLAVE 3** U370 <28> Mask Register and bit 3 of **Interrupt Controllers MASTER** U350 <28> Mask Register high.

Error Index 12 The **Timer** COUNTER 1 interrupt did not occur or it could not be cleared at the **Interrupt Controllers SLAVE 3** U370 <28>.

Routine Name Timer 2 Interrupt

Overview This test verifies the **Timer U822 <25> COUNTER 2** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Configure the **Timer COUNTER 2** to count 6 MHz clock.
 - Write *00hex* to **Timer Configuration Logic U720 <25>** so that G2 U822-16 <25> is high and PCLK(H) is connected to CLK2 U822-18 <25>.
2. Clear **Timer COUNTER 2** interrupt.
 - Write *B0hex* to the Control Word Register.
3. Verify that there is no **Timer COUNTER 2** interrupt pending at the **Interrupt Controllers SLAVE 3 U370 <28>**.
 - Read **Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER** and verify that IR0 U370-19 is low.
4. If there is no **Timer COUNTER 2** interrupt pending, then enable the **Timer COUNTER 2** interrupt by setting bit 1 of **Interrupt Controllers SLAVE 3 U370 <28> Mask Register** and bit 3 of **Interrupt Controllers MASTER U350 <28> Mask Register** low.
5. Complete programming the **Timer COUNTER 2** with *80hex* as the starting count.
 - Write *80hex* and *00hex* to the Count Register low and high bytes, respectively. This starts the counter.
6. Perform a software delay.
7. Verify that **Timer COUNTER 2** interrupt did occur.
8. Disable the **Timer COUNTER 2** interrupt by setting bit 1 of **Interrupt Controllers SLAVE 3 U370 <28> Mask Register** and bit 3 of **Interrupt Controllers MASTER U350 <28> Mask Register** high.

Error Index 11 The **Timer COUNTER 2** interrupt did not occur or it could not be cleared at the **Interrupt Controllers SLAVE 3 U370 <28>**.

Routine Name 80287 Math Coprocessor

Overview This test verifies the floating point and transcendental function capabilities of the **Numeric Processor Extension U500 <27>**.

Description 1. Compute the following equation using the **Numeric Processor Extension U500 <27>** and verify that the result is within a tolerance of ± 0.005 of the expected value.

$$\frac{\log_e(15.3) * \log_{10}(23.5) * \exp(0.12)}{\text{atan}[\sin(0.7071) * \cos(0.5) * \tan(0.5774)]}$$

Error Index 10 The result from the floating point calculation was not 11.8582 ± 0.005 .

Routine Name DMA Controller Interrupt

Overview This test verifies the **DMA Controller <28>** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated by performing a memory-to-memory DMA transfer.

Description

1. Check to see if the **DMA Controller** is accessible.
 - Write, read, and verify *7F01hex* to the General Mode Register of the **DMA Controller**.
2. If the **DMA Controller** is accessible (i.e., *7F01hex* is verified in the step 1), program the **DMA Controller** channel 0 and clear its interrupt, if any.
 - Write *7B01hex*, *00hex*, *00hex*, *0013hex*, *0000hex*, and *0018hex* to General Mode Register, General Burst Register, General Delay Register, Command Pointer Register High, Command Pointer Register Low and General Command Register, respectively.
3. Verify that there is no DMA interrupt pending at the **Interrupt Controllers SLAVE 1 U360 <28>**.
 - Read **Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER** and verify that **IR6 U360-24** is low.
4. If there is no DMA interrupt pending, construct the command block for the transfer at **DRAM** address *30000hex*. Initialize the source and destination memory locations.
 - Write *C8DDhex*, *0013hex*, *0020hex*, *0013hex*, *0040hex*, *0000hex*, *0010hex*, *0000hex*, and *0401hex* to locations starting from *30000hex*.
 - Initialize the source **DRAM** address range *30020hex*-*3002Ahex* with patterns *AAAAhex*, *CCCChex*, *F0F0hex*, *FF00hex*, and *5555hex*.
 - Initialize the destination **DRAM** address range *30040hex*-*30049hex* with pattern *8000hex*.
5. Enable the DMA interrupt by setting bit 6 of **Interrupt Controllers SLAVE 1 U360 <28> Mask Register** low and bit 1 of **Interrupt Controllers MASTER U350 <28> Mask Register** low.
6. Start the DMA transfer by writing *1Ahex* to General Command Register.
7. Perform a software delay.
8. Verify that the DMA interrupt occurred.
9. Disable the DMA interrupt by setting bit 6 of **Interrupt Controllers SLAVE 1 U360 <28> Mask Register** high and bit 1 of **Interrupt Controllers MASTER U350 <28> Mask Register** high.

Error Index 0F

The value read from the DMA Controller's General Mode Register did not match what was written; or the DMA interrupt did not occur; or the DMA interrupt could not be cleared at **Interrupt Controllers SLAVE 1 U360**.

Routine Name Front Panel Controller Interrupt

Overview This test verifies the **Front Panel Controller <3>** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The infrared LEDs are disabled and enabled to generate the interrupt.

Description

1. Initialize the **Front Panel Controller <3>**.
 - Write *04hex* and *22hex* to the Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
2. Verify that there is no front panel interrupt pending at the **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that IR4 U350-22 <28> is low.
3. If there was no pending front panel interrupt, enable the front panel interrupt by setting the **Interrupt Controllers MASTER U350 <28> Mask Register** bit 4 low.
4. Disable the front panel infrared LEDs by writing *00hex* to **Front Panel Infrared Disable Control U700B <26>** so that IRDIS(L) U212C-11(L) <3> is low.
5. Perform a software delay.
6. Enable the front panel infrared LEDs by writing *01hex* to **Front Panel Infrared Disable Control U700B <26>** so that IRDIS(L) U212C-11(L) <3> is high.
7. Perform a software delay.
8. Verify that the front panel interrupt did occur.
9. Disable the front panel interrupt by setting **Interrupt Controllers MASTER U350 Mask Register** bit 4 high.
10. Initialize the **Front Panel Controller**.
 - Write *04hex* and *22hex* to the Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.

Error Index 0E The **Front Panel Controller <3>** interrupt did not occur or it could not be cleared at the **Interrupt Controllers MASTER U350 <28>**.

Routine Name Serial Data Interface Interrupt

Overview This test verifies the **Serial Data Interface** SDI IC U330 <25> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program the **Serial Data Interface** SDI IC U330 <25> to loopback mode.
 - Set bits 3 and 4 (Receiver Reset and Transmitter Reset bits) of the Miscellaneous Control Select 1 latch high.
 - Write *38hex* and *00hex* to the SDI Control Select Register and SDI Interrupt Output Mask Register.
 - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.
2. Verify that there is no SDI interrupt pending at the **Interrupt Controllers** <28>.
 - Read **Interrupt Controllers** SLAVE 1 U360 <25> INTERRUPT REQUEST REGISTER and verify that IR4 U360-22 is low.
3. If there is no pending interrupt, enable the SDI interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 4 low.
4. Enable the transmitter empty interrupt at the **Serial Data Interface** SDI IC U330 <25> by writing *01hex* to the SDI Interrupt Output Mask Register.
5. Perform a software delay and then verify that the SDI interrupt did occur.
6. Disable the SDI interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 4 high.
7. Program the **Serial Data Interface** SDI IC U330 <25> to normal mode.
 - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch high.
 - Write *3Chex* to the SDI Control Select Register.
 - Read SDI I/O Status Register to find out the state of Left Data Back (bit 5), Center Data Back (bit 3), and Right Data Back (bit 0) bits.

- Write a value to the SDI Control Select Register so that bit 2 is high and bits 3, 4, and 5 reflect the state of Left Data Back, Center Data Back, and Right Data Back, respectively.
- Write 0E_{hex} to SDI Interrupt Output Mask Register.
- Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.

Error Index 0D

The SDI interrupt did not occur or it could not be cleared at the **Interrupt Controllers SLAVE 1 U360 <28>**.

Routine Name Real Time Clock Interrupt

Overview This test verifies the **Real Time Clock <25>** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Mask all the **Real Time Clock <25>** interrupts by writing *00hex* to its Mask Register.
2. Read the Interrupt Status Register of the **Real Time Clock <25>** to clear all the interrupts.
3. Verify that there is no **Real Time Clock** interrupt at the **Interrupt Controllers <28>**.
 - Read the **Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER** and verify that **IR6 U370-24** is low.
4. If there is no pending **Real Time Clock** interrupt at the **Interrupt Controllers** (i.e., **IR6 U370-24** is low), enable the interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers SLAVE 3 U370 <28> Mask Register bit 6** and **Interrupt Controllers MASTER U350 <28> Mask Register bit 3** low.
5. Enable the **Real Time Clock's** one hundredth interrupt at the **Real Time Clock <25>** by writing *18hex* and *02hex* to the Command Register and Mask Register, respectively.
6. Perform a software delay (at least 10 ms).
7. Disable the **Real Time Clock's** one hundredth interrupt at the **Real Time Clock <25>** by writing *08hex* to the Command Register.
8. Verify that the **Real Time Clock's** interrupt did occur.
9. Disable **Real Time Clock's** interrupt at the **Interrupt Controllers** by setting **Interrupt Controllers SLAVE 3 U370 <28> bit 6** and **Interrupt Controllers MASTER U350 <28> bit 3** high.

Error Index 0C The **Real Time Clock** interrupt did not occur or it could not be reset at the **Interrupt Controllers Slave U370 <28>**.

Routine Name GPIB Controller Interrupt

Overview This test verifies that the GPIB interrupt can be cleared by resetting the GPIB Controller U410 <5>.

Description

1. Reset GPIB Controller U410 <5> and verify its reset status.
 - Write *80hex*, *00hex*, *00hex*, and *00hex* to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.
 - Read Interrupt Status Register 0 and Interrupt Status Register 1 and verify that they both read *00hex*.
2. Verify that there is no GPIB interrupt pending at the Interrupt Controllers <28>.
 - Read Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER and verify that IR2 U370-20 is low.

Error Index 0B The GPIB Controller <5> reset did not set the Interrupt Status Register 0 to *00hex*; or the GPIB Controller <5> reset did not set the Interrupt Status Register 1 to *00hex*; or the GPIB Controller <5> interrupt could not be reset at the Interrupt Controllers SLAVE 3 U370 <28>.

Routine Name Printer Controller Interrupt

Overview This test verifies the **Printer Controller U430 <6>** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The interrupt is generated by looping back a byte of data.

Description

1. Program the **Printer Controller <6>** in test mode (loop back mode) by writing *A6hex* and *08hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is low. This enables **Printer Loop Back Buffer U540 <6>** and disables **Printer Interface Control buffer U541 <6>**.
2. Verify that there is no printer interrupt pending at the **Interrupt Controllers <28>**.
 - Read the **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that IR6 U350-24 is low.
3. If there are no pending interrupts, enable the interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register** bit 6 low.
4. Enable the interrupt at the **Printer Controller** by writing *0Dhex* to the Control Word Register.
5. Loop back a pattern.
 - Write *AAhex* to U430 Port A, perform a software delay, and read it from U430 Port B. Immediately following the read, a transmitter empty interrupt is generated, i.e., INTR(A)(H) U430-17 is high.
6. Perform a software delay.
7. Disable the interrupt at the **Printer Controller <6>** by writing *0Chex* to the Control Word Register.
8. Disable the interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register** bit 6 high.
9. Verify that a printer interrupt did occur.
10. Program the **Printer Controller <6>** in normal mode by writing *09hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is high.

Error Index 0A The **Printer Controller <6>** interrupt did not occur or it could not be cleared at the **Interrupt Controllers MASTER U350 <28>**.

Routine Name Std RS-232 Controller Interrupt

Overview This test verifies the Std RS-232 Controller's U311 <5> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program Std RS-232 Controller U311 <5> in local loop back mode with eight bits per character, one stop bit, no parity and 9600 baud rate.
 - Write 35hex, 25hex, 1Ahex, 13hex, 87hex, BBhex, and E5hex to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
 - Read IPCR, write 00hex to IMR, and read ISR.
 - Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Verify that there is no pending RS-232 interrupt at the **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER** and verify that IR4 U350-22 is low.
3. If there is no pending RS-232 interrupt, enable the RS-232 interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 3** and **Interrupt Controllers SLAVE 3 U370 Mask Register bit 4** low.
4. Enable the receiver ready interrupt at the Std RS-232 Controller U311 <5> by writing 02hex to IMR.
5. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
6. Irrespective of the transmitter availability, transmit a character by writing 55hex to THRA.
7. Perform a software delay.
8. Read the character by reading RHRA.
9. Verify that the interrupt did occur.
10. Reset the receiver and disable the receiver interrupt by writing 25hex and 00hex to CRA and IMR, respectively.
11. Disable the RS-232 interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 3** and **Interrupt Controllers SLAVE 3 U370 <28> Mask Register bit 4** high.
12. Program the Std RS-232 controller U311 <5> in normal mode with eight bits per character, one stop bit, no parity and 9600 baud rate.

- Write *1Ahex*, *13hex*, *07hex*, *BBhex*, and *E5hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
- Read IPCR, write *00hex* to IMR, and read ISR.
- Write *01hex*, *04hex*, *00hex*, *06hex*, *30hex*, *20hex*, and *05hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index 09

The Std RS-232 Controller <5> interrupt did not occur or it could not be reset at the Interrupt Controllers SLAVE U370 <28>.

Routine Name MMU Display Talk Request Interrupt

Overview This test verifies the MMU Gate Array U210 <33> display talk request interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the MMU Gate Array U210 <33> to an isolated diagnostic mode by writing *1Ahex*, *007Fhex*, and *4hex* to **Diagnostics U524 <33>**, **U210 Status and Mode Register (SMR)**, and **Diagnostics U530 <33>**, respectively.
2. Verify that there is no pending display talk request interrupt at the **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER** and verify that IR2 U360-20 is low.
3. If there is no display interrupt pending, then enable display talk request interrupt by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 2** low.
4. Generate a display talk request interrupt by setting **DATARDY(H) U524-6 <33>** high and then low.
5. Perform a software delay and then verify that the display talk request interrupt did occur.
6. Clear the interrupt at the MMU by setting **SMR bit 1** high.
7. Disable the display talk request interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 2** high.
8. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to **Diagnostics U530 <33>** and **U210 Status and Mode Register**.

Error Index 08 The display talk request interrupt did not occur or it could not be cleared at the **Interrupt Controllers SLAVE 1 U360 <28>**.

Routine Name MMU SAG Interrupt

Overview This test verifies the MMU Gate Array U210 <33> SAG Interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the MMU Gate Array U210 <33> to an isolated diagnostic mode by writing *1Ahex*, *007Fhex*, and *4hex* to **Diagnostics U524 <33>**, **U210 Status and Mode Register (SMR)**, and **Diagnostics U530 <33>**, respectively.
2. Verify that there is no pending SAG interrupt at the **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER** and verify that **IR0 U360-18** is low.
3. If there is no SAG interrupt pending, enable the SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0** low.
4. Start a "write display" cycle by setting bit 9 of SMR high and setting **SENDNEW(L) U524-5 <33>** low and then high.
5. Perform a software delay and then verify that the SAG interrupt did occur.
6. Clear SAG interrupt at the MMU by setting bit 3 of SMR high.
8. Disable the SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0** high.
9. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to **Diagnostics U530 <33>** and **U210 Status and Mode Register**.

Error Index 07 The SAG interrupt did not occur or it could not be cleared at the **Interrupt Controllers SLAVE 1 U360 <28>**.

Routine Name MMU RAG Interrupt

Overview This test verifies MMU Gate Array's U210 <33> RAG interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the MMU Gate Array U210 <33> to an isolated diagnostic mode by writing *1Ahex*, *007Fhex*, and *4hex* to **Diagnostics U524 <33>**, **U210 Status and Mode Register (SMR)**, and **Diagnostics U530 <33>**, respectively.
2. Verify that there is no RAG interrupt pending at **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER** and verify that **IR1 U360-19** is low.
3. If there is no RAG interrupt pending, enable the RAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 1** low.
4. Generate an "end of digitizer message" interrupt.
 - Set **DIGREQ(L) U524-12 <33>** low and then high with **U524-19(H):H <33>**.
 - Set **DIGREQ(L) U524-12 <33>** high with **U524-19(H):L <33>**.
5. Perform a software delay and then verify that the RAG interrupt did occur.
6. Clear the RAG interrupt at the MMU by setting bit 4 of the **Status and Mode Register (SMR)** high.
7. Disable RAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 1** high.
8. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to **Diagnostics U530 <33>** and **U210 Status and Mode Register**.

Error Index 06 The RAG interrupt did not occur or it could not be cleared at the **Interrupt Controllers SLAVE U360 <28>**.

Routine Name DMA 07

Overview This test the DMA Controller <28> channel 0 by performing a memory DMA transfer in DRAM <30>.

Description

1. Check if the DMA Controller is accessible.
 - Read, and verify 7F01hex to the General Mode Register of the Controller.
2. If Controller is accessible (i.e., 7F01hex was verified in step 1), the command block for the transfer at DRAM <30> address initialize the source and destination memory locations.

C8DDhex, 0013hex, 0020hex, 0013hex, 0040hex, 0000hex, 0010hex, 0000hex, and 0401hex to locations starting from 30000hex.

Initialize the source DRAM address range 30020hex-3002Ahex with patterns AAAAhex, CCCChex, F0F0hex, FF00hex, and 5555hex.

Initialize the destination DRAM address range 30040hex-30049hex with pattern 8000hex.
3. Write the DMA Controller channel 0 to read the command block from DRAM <30> and start the transfer of five words from 30020hex to 30040hex.

Write 7F01hex, 00hex, 00hex, 0013hex, 0000hex, and 001Ahex to General Mode Register, General Burst Register, General Delay Register, General Command Pointer Register High, Command Pointer Register Low, and General Command Register, respectively.
4. Insert software delay.
5. DMA interrupt at the DMA Controller and verify that the transfer was successful.

Read 0018hex General Command Register.

Compare the contents of the destination and source memory locations and verify that they are the same.

Error Index 05

The DMA Controller's General Mode Register did not contain 7F01hex; the DMA Controller was not written; or the DMA transfer was not successful.

Routine Name Std RS-232 External Loopback

Overview This test verifies the Std RS-232 Controller U311 <5>, RS-232 Clock Generator <5> and Std RS-232 Interface Drivers <5> by externally looping back a set of patterns. Its use is only foreseen in situations where no diagnostic menus can be displayed on the screen (i.e., the operator needs to, or would like to, use the external diagnostic RS-232 terminal mode), yet the RS-232 port also contains a fault. In other words, there are multiple, non-kernel type faults in the instrument which prohibit being able to obtain diagnostic screen information.

Operator Procedure This test only executes when the lower straps of S710 on the I/O board are set to 01bin (bottom strap to the right, the one above it set to the the left). This forces the test to be repeatedly executed after all other Executive Kernel tests have successfully run. The operator should connect an external loopback connector (Tek part no. 013-0198-00) to the RS232 connector on the rear panel of the instrument. External test equipment must be used to diagnose faults.

Description

1. Program the Std RS-232 Controller U311 <5> in normal mode with eight bits per character, one stop bit, no parity and 9600 baud rate.
 - Write 1Ahex, 13hex, 07hex, BBhex, and E5hex to CRA, MR1A, MR2A, CSR, and ACR, respectively.
 - Read IPCR, write 00hex to IMR, and read ISR.
 - Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
3. Irrespective of the transmitter availability, transmit pattern AAhex by writing it to THRA.
4. Wait for the receiver to be ready by repeatedly reading SRA until bit 0 is high or until a software timeout period has expired.
5. Irrespective of the receiver availability, receive the pattern by reading RHRA and verify it.
6. Repeat steps 2-5 for patterns CChex, F0hex, 0Fhex, and 55hex.
7. Program the Std RS-232 Controller U311 <5> in normal mode with eight bits per character, one stop bit, no parity and 9600 baud rate.
 - Write 1Ahex, 13hex, 07hex, BBhex, and E5hex to CRA, MR1A, MR2A, CSR, and ACR, respectively.
 - Read IPCR, write 00hex to IMR, and read ISR.

- Write *01hex*, *04hex*, *00hex*, *06hex*, *30hex*, *20hex*, and *05hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index 04

Not applicable since this test is user selected.

Display Kernel

RAM Data Lines (FF)

Routine Name	RAM Data Lines
Overview	This test verifies the data lines to General Purpose Static RAM <39> by performing a "walking one's" test on General Purpose Static RAM address 00000hex.
Description	<ol style="list-style-type: none">1. Perform a "walking one's" test on General Purpose Static RAM address 00000hex.<ul style="list-style-type: none">• Write the pattern 8000hex to address 00000hex. Read the same address and verify that the pattern was 8000hex. Continue this write/read/verify sequence with patterns 4000hex, 2000hex, 1000hex, 0800hex, 0400hex, 0200hex, 0100hex, 0080hex, 0040hex, 0020hex, 0010hex, 0008hex, 0004hex, 0002hex, 0001hex, and 0000hex.
Error Index FF	The pattern read from General Purpose Static RAM <39> address 00000hex was not the pattern written.

Routine Name RAM Address/Data

Overview This test verifies the address lines and data integrity of **General Purpose Static RAM** <39> address range 00000-00FFF_{hex} by performing a RAM test on this address range.

Description

1. Verify address range 00000-00FFF_{hex}. Terminate test if any verify operation fails.
 - Fill address range 00000-00FFF_{hex} with the pattern AAAA_{hex}.
 - Read and verify address 00000_{hex} for AAAA_{hex}. If so, write CCCC_{hex} to address 00000_{hex}. Increment address and continue this read/verify/write sequence until address 00FFF_{hex} is reached.
 - Repeat the read/verify/write sequence, starting again at address 00000_{hex} for CCCC_{hex}, F0F0_{hex}, 5555_{hex}, AAAA_{hex}, and 0000_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index FE The pattern read from a **General Purpose Static RAM** address location was not the pattern written. On failure, the RAM Data Lines test (FF) is repeatedly performed on the failed address location.

Display Kernel

ROM U612 Location (FD)

Routine Name	ROM U612 Location
Overview	This test verifies that ROM & Select U612 <39> is in the correct socket.
Description	<ol style="list-style-type: none">1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.<ul style="list-style-type: none">• Exclusive-or the contents of byte locations E0008_{hex} and E000A_{hex} and verify that the result is FF_{hex}.2. If the complement test was successful, then verify the location byte of the EPROM device.<ul style="list-style-type: none">• Read and verify the location byte against the known value.
Error Index FD	The bytes at E0008 _{hex} and E000A _{hex} were not complementary or the location byte did not match the known value.

Display Kernel

ROM U602 Location (FC)

Routine Name ROM U602 Location

Overview This test verifies that **ROM & Select U602 <39>** is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations *E0009hex* and *E000Bhex* and verify that the result is *FFhex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.

Error Index FC The bytes at *E0009hex* and *E000Bhex* were not complementary or the location byte did not match the known value.

Routine Name ROM U612 Checksum

Overview This test verifies the integrity of ROM & Select U612 <39> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0008hex and E000Ahex and verify that the result is FFhex.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U612 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index FB The bytes at E0008hex and E000Ahex were not complementary or the computed checksum did not match the stored checksum.

Routine Name ROM U602 Checksum

Overview This test verifies the integrity of ROM & Select U602 <39> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations *E0009hex* and *E000Bhex* and verify that the result is *FFhex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U602 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index FA The bytes at *E0009hex* and *E000Bhex* were not complementary or the computed checksum did not match the stored checksum.

Routine Name DMA 0

Overview This test verifies CPU & Ready Logic U524 <39> Programmable DMA Unit's channel 0 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 0 interrupt by writing 0004*hex* to DMA 0 Interrupt Control Register.
2. Initialize the source and destination memory locations in **General Purpose Static RAM** <39>.
 - Write patterns AAAA*hex*, CCC*hex*, F0F0*hex*, FF00*hex* and 5555*hex* to source memory locations starting at address 210*hex*.
 - Write patterns 5555*hex*, 3333*hex*, 0F0F*hex*, 00FF*hex* and AAAA*hex* to destination memory locations starting at address 200*hex*.
3. Program the DMA channel 0 to transfer the five source patterns to the destination memory locations starting at 200*hex* and generate an interrupt after completion of the transfer.
 - Write B725*hex*, 0005*hex*, 0000*hex*, 0200*hex*, 0000*hex*, 0210*hex* and B727*hex* to DMA 0 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source pointer Lower, and DMA 0 Channel Control Word registers, respectively. The last write initiates the DMA transfer.
4. Disable DMA 0 interrupt by writing 000C*hex* to DMA 0 Interrupt Control Register.
5. Read and verify the destination memory locations.

Error Index F9 One or more of the transferred patterns did not match the expected value.

Routine Name DMA 1

Overview This test verifies CPU & Ready Logic U524 <39> Programmable DMA Unit's channel 1 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 1 interrupt by writing 0004hex to DMA 1 Interrupt Control Register.
2. Initialize the source and destination memory locations in General Purpose Static RAM <39>.
 - Write patterns AAAAhex, CCCChex, F0F0hex, FF00hex and 5555hex to source memory locations starting at address 210hex.
 - Write patterns 5555hex, 3333hex, 0F0Fhex, 00FFhex and AAAAhex to destination memory locations starting at address 200hex.
3. Program the DMA channel 1 to transfer the five source patterns to the destination memory locations starting at 200hex and generate an interrupt after completion of the transfer.
 - Write B725hex, 0005hex, 0000hex, 0200hex, 0000hex, 0210hex, and B727hex to DMA 1 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source pointer Lower, and DMA 1 Channel Control Word registers, respectively. The last write initiates the DMA transfer.
4. Disable DMA 1 interrupt by writing 000Chex to DMA 1 Interrupt Control Register.
5. Read and verify the destination memory locations.

Error Index F8 One or more of the transferred patterns did not match the expected value.

Routine Name Timer 0

Overview This test verifies CPU & Ready Logic U524 <39> Programmable Timers' timer 0 counting accuracy and its interrupt. The counting accuracy of the timer 0 is verified by counting the system clock for a short duration. If the timer 0 counts accurately, a timer 0 interrupt is generated and verified.

Description

1. Enable the timer 0 to count the system clock by setting its gate signal high.
 - Set Diagnostic Control/Status Latch TMR IN 0 <43> high by writing 83hex to U600 <43>.
2. Stop the timer 0 by writing 4000hex to its Timer Mode/Control Register.
3. Program the counter to count up by writing 0000hex, FFFFhex, and C000hex to timer 0 Count Register, Max Count Value A Register, and Timer 0 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing 4000hex to its Timer Mode/Control Register.
6. Read and verify the timer 0 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 0 is counting accurately), enable the timer 0 to count the system clock again by setting its gate signal high. If not, terminate testing.
 - Set Diagnostic Control/Status Latch TMR IN 0 <43> high by writing 83hex to U600 <43>.
8. Enable timer 0 interrupt by writing 0004hex to the Timer 0 Interrupt Control Register.
9. Program the counter to count up to 100hex and generate an interrupt when the count reaches 100hex.
 - Write 0100hex, 0000hex, and E000hex to timer 0 Max Count Value A Register, Count Register, and Timer 0 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing 4000hex to its Timer Mode/Control Register.
12. Disable timer 0 interrupt by writing 000Chex to the Timer 0 Interrupt Control Register.
13. Verify that the timer 0 interrupt did occur.

Display Kernel

Timer 0 (F7)

Error Index F7

The timer 0 count was not within the expected tolerance or the timer 0 interrupt did not occur.

Routine Name Timer 1

Overview This test verifies CPU & Ready Logic U524 <39> Programmable Timers' timer 1 counting accuracy and its interrupt. The counting accuracy of the timer 1 is verified by counting the system clock for a short duration. If the timer 1 counts accurately, a timer 1 interrupt is generated and verified.

Description

1. Enable the timer 1 to count the system clock by setting its gate signal high.
 - Set Diagnostic Control/Status Latch TMR IN 1 <43> high by writing *A3hex* to U600 <43>.
2. Stop the timer 1 by writing *4000hex* to its Timer Mode/Control Register.
3. Program the counter to count up by writing *0000hex*, *FFFFhex*, and *C000hex* to timer 1 Count Register, Max Count Value A Register, and Timer 1 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing *4000hex* to its Timer Mode/Control Register.
6. Read and verify the timer 1 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 1 is counting accurately), enable the timer 1 to count the system clock again by setting its gate signal high. If not, terminate testing.
 - Set Diagnostic Control/Status Latch TMR IN 1 <43> high by writing *A3hex* to U600 <43>.
8. Enable timer 1 interrupt by writing *0004hex* to the Timer 1 Interrupt Control Register.
9. Program the counter to count up to *100hex* and generate an interrupt when the count reaches *100hex*.
 - Write *0100hex*, *0000hex*, and *E000hex* to timer 1 Max Count Value A Register, Count Register, and Timer 1 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing *4000hex* to its Timer Mode/Control Register.
12. Disable timer 1 interrupt by writing *000Chex* to the Timer 1 Interrupt Control Register.
13. Verify that the timer 1 interrupt did occur.

Display Kernel

Timer 1 (F6)

Error Index F6

The timer 1 count was not within the expected tolerance or the timer 1 interrupt did not occur.

Routine Name Timer 2

Overview This test verifies CPU & Ready Logic U524 <39> Programmable Timers' timer 2 counting accuracy and its interrupt. The counting accuracy of the timer 2 is verified by counting the system clock for a short duration. If the timer 2 counts accurately, a timer 2 interrupt is generated and verified.

Description

1. Stop the timer 2 by writing 4000*hex* to its Timer Mode/Control Register.
2. Program the counter to count up by writing 0000*hex*, FFFF*hex*, and C000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
3. Perform a software delay.
4. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
5. Read and verify the timer 2 count against the known value for the expected tolerance.
6. If the count was within the expected tolerance (i.e., the timer 2 is counting accurately), enable the timer 2 interrupt by writing 0004*hex* to the Timer 2 Interrupt Control Register. If not, terminate testing.
7. Program the counter to count up to 100*hex* and generate an interrupt when the count reaches 100*hex*.
 - Write 0100*hex*, 0000*hex*, and E000*hex* to timer 2 Max Count Value A Register, Count Register, and Timer 2 Mode/Control Register, respectively.
8. Perform a software delay.
9. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.
10. Disable timer 2 interrupt by writing 000C*hex* to the Timer 2 Interrupt Control Register.
11. Verify that the timer 2 interrupt did occur.

Error Index F5 The timer 2 count was not within the expected tolerance or the timer 2 interrupt did not occur.

Routine Name Executive Communication

Overview This test verifies the communication path from the Display processor to the Executive processor, through the **Interface Data Buffers <43>** and the **Executive Processor Parallel Interface Port <43>**, by inverting and echoing a specific set of "walking ones" and "walking zeros" patterns which are expected from the Executive processor. If, at any time, the Display processor receives an unexpected pattern, it enters an infinite loop in which it inverts and echo's every pattern sent thereafter.

Description

1. Set up CPU & Ready Logic 80186 DMA 0<39> to receive a data pattern from the **Executive Processor Parallel Interface Port <43>** and write it to **General Purpose Static RAM <39>** memory location 00210_{hex} without generating any interrupt at the end of the transfer.
2. Set up CPU & Ready Logic 80186 DMA 1<39> to send a data pattern from **General Purpose Static RAM <39>** memory location 00210_{hex} to the **Executive Processor Parallel Interface Port <43>** without generating any interrupt at the end of the transfer.
3. Initialize the **Executive Processor Parallel Interface Port <43>** by reading port latches U731 & U733 through **Interface Data Buffers <43>**.
4. Wait for a data pattern to be received from the Executive processor through DMA 0.
5. If the received data pattern matches the expected pattern, then invert all bits (i.e., 1's complement) and allow DMA 1 to send the inverted pattern back to the Executive processor (see Error Index section below).
6. If the received data pattern did not match the expected pattern, perform an infinite loop in which any patterns received, as in step 4, are inverted and sent back to the Executive processor, as in step 5.
7. Repeat steps 4 & 5 until all expected patterns have been received, inverted, and sent back (see Error Index section below).

Error Index F4 One of the data patterns received from the Executive processor did not match the expected pattern. The expected patterns to be received and those returned to the Executive processor are shown in the following table:

Display Kernel

Executive Communication (F4)

Expected Pattern Received (<i>hex</i>)	Pattern Sent (<i>hex</i>)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	EFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
EFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001

Routine Name CPU Registers

Overview This test verifies the CPU U440 <17> registers.

Description

1. Initialize the chip select registers.
2. Test the processors general purpose registers.
 - Transfer the test pattern 5AA5_{hex} between the registers in the following order: ax, ds, bx, ss, cx, es, dx, di, bp, sp, si. Compare the contents of si to ax.
 - Transfer the test pattern A55A_{hex} between the registers in the following order: ax, ds, bx, ss, cx, es, dx, di, bp, sp, si. Compare the contents of si to ax.
 - Transfer zero to the low order eight bits of the flags register and execute the following instructions: jb, jp, je, js. If any of these instructions branch there is a flags register failure.
 - Transfer FFFF_{hex} to the low order eight bits of the flags register and execute the following instructions: jnb, jnp, jne, jns. If any of these instructions branch there is a flags register failure.
3. Read and verify the chip select registers.

Error Index 1 One of the above listed registers did not verify correctly. After displaying the error code on the LEDs, the processor halts.

Routine Name ROM U400 Location/Checksum

Overview This test verifies the location and integrity of ROM & Select U400 <17> by performing a complement, location, and checksum test.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0008_{hex} and E000A_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.
3. If the location test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U400 except the first two, and then verify it against the checksum stored in the first two bytes of the device.

Error Index 2 The bytes at E0008_{hex} and E000A_{hex} were not complementary; or the location byte at E000C_{hex} did not match the expected value; or the checksum for the device was not correct. After displaying the error code on the LEDs, the processor halts.

Routine Name	ROM U410 Location/Checksum
Overview	This test verifies the location and integrity of ROM & Select U410 <17> by performing a complement, location, and checksum test.
Description	<ol style="list-style-type: none">1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.<ul style="list-style-type: none">• Exclusive-or the contents of byte locations E0009_{hex} and E000B_{hex} and verify that the result is FF_{hex}.2. If the complement test was successful, then verify the location byte of the EPROM device.<ul style="list-style-type: none">• Read and verify the location byte against the known value.3. If the location test was successful, then perform a checksum on the contents of the EPROM device.<ul style="list-style-type: none">• Perform a checksum on all the bytes in U410 except the first two, and then verify it against the checksum stored in the first two bytes of the device.
Error Index 3	The bytes at E0009 _{hex} and E000B _{hex} were not complementary; or the location byte at E000D _{hex} did not match the expected value; or the checksum for the device was not correct. After displaying the error code on the LEDs, the processor halts.

Routine Name

Static RAM Data Lines

Overview

This test verifies **Ram Select** <17> and the data lines from CPU U440 <17>, through the **µP Kernel Data Buffers** <17>, to the **General Purpose Static RAM** <17> by performing a "walking one's" test on two static RAM memory locations.

Description

1. Initialize the test locations.
 - Write the pattern *FFFFhex* to address *00000hex* and *10000hex*.
2. Verify **Ram Select** <17>.
 - Read address *00000hex* and verify that it is *FFFFhex*.
3. Perform a "walking one's" test on **General Purpose Static RAM** <17> address *00000hex*. Terminate test if any verify operation fails.
 - Write the pattern *8000hex* to address *00000hex*. Read the same address and verify that it was *8000hex*. Continue this write/read/verify sequence with the patterns *4000hex*, *2000hex*, *1000hex*, *0800hex*, *0400hex*, *0200hex*, *0100hex*, *0080hex*, *0040hex*, *0020hex*, *0010hex*, *0008hex*, *0004hex*, *0002hex*, *0001hex*.
4. Verify **Ram Select** <17>.
 - Write *0000hex* to address *00000hex*.
 - Read address *10000hex* and verify that it is *FFFFhex*.
5. Perform a "walking one's" test on **General Purpose Static RAM** <17> address *10000hex*. Terminate test if any verify operation fails.
 - Write the pattern *8000hex* to address *10000hex*. Read the same address and verify that it was *8000hex*. Continue this write/read/verify sequence with the patterns *4000hex*, *2000hex*, *1000hex*, *0800hex*, *0400hex*, *0200hex*, *0100hex*, *0080hex*, *0040hex*, *0020hex*, *0010hex*, *0008hex*, *0004hex*, *0002hex*, *0001hex*.
6. Verify **Ram Select** <17>.
 - Write *0000hex* to address *10000hex*.
 - Read address *00000hex* and address *10000hex* and verify that both are *0000hex*.

Error Index 4

The patterns read from **General Purpose Static RAM** <17> address *00000hex* or address *10000hex* was not the same pattern written. The test will endlessly write and read the failing test pattern to the failing memory location as a debugging aide.

Routine Name	Static RAM Address/Data
Overview	This test verifies the address lines and data integrity of the General Purpose Static RAM <17> by performing a RAM test on the first 16k bytes of static RAM memory.
Description	<ol style="list-style-type: none">1. Verify General Purpose Static RAM <17> address range 00000_{hex} to 13FFF_{hex}. Terminate test if any verify operation fails.<ul style="list-style-type: none">• Fill address range 00000_{hex} to 13FFF_{hex} with the pattern AAAA_{hex}.• Read and verify address 00000_{hex} for AAAA_{hex}. If so, write CCCC_{hex} to address 00000_{hex}. Increment the address and continue this read/verify/write sequence until address 13FFF_{hex} is reached.• Repeat the read/verify/write sequence, starting again at address 00000_{hex}, for F0F0_{hex}, 5555_{hex}, and AAAA_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern).• Make one last read/verify pass for the pattern AAAA_{hex} starting at address 00000_{hex}.
Error Index 5	The pattern read from a memory location in the General Purpose Static RAM <17> was not the pattern written. The test will endlessly write and read the failing test pattern to the failing memory location as a debugging aide.

Routine Name Processor Peripherals

Overview This test verifies **Programmable Timers U440 <17>** timer 2's counting accuracy, **Programmable DMA Unit's U440 <17>** channel 0 and channel 1, and the **Programmable Interrupt Controller U440 <17>**. The counting accuracy of timer 2 is verified by counting the system clock for a short duration. The DMA channels are verified by transferring a set of patterns from one group of memory locations (source) to another (destination). The Interrupt Controller is verified by testing its registers and by generating internal interrupts.

Description PROGRAMMABLE TIMERS

1. Stop the timer 2 by writing `4000hex` to its Timer Mode/Control Register.
2. Program the counter to count up by writing `0000hex`, `FFFFhex`, and `C000hex` to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.
3. Perform a software delay.
4. Stop the timer by writing `4000hex` to its Timer Mode/Control Register.
5. Read and verify the timer 2 count against the known value for the expected tolerance.

PROGRAMMABLE DMA UNIT

6. Set 35 destination memory locations to `0000hex`.
7. Program DMA channel 0 to transfer 35 source patterns to the destination memory.
 - Write `B725hex` and `0023hex` to DMA 0 Channel Control Word and Transfer Count registers, respectively. The address values written to the Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, and Source Pointer Lower registers are determined at test execution time. Write `B727hex` to the DMA 0 Channel Control Word to initiate the DMA transfer.
8. Wait 1 ms for DMA 0 to finish the transfer.
9. Read and verify that the Transfer Count Register is zero.
10. Read and verify the destination memory locations.
11. Set the 35 destination memory locations to `0000hex`.
12. Program the DMA channel 1 to transfer the 35 source patterns to the destination memory.

- Write *B725hex* and *0023hex* to DMA 1 Channel Control Word and Transfer Count registers, respectively. The address values written to the Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, and Source Pointer Lower registers are determined at test execution time. Write *B727hex* to the DMA 1 Channel Control Word to initiate the DMA transfer.

13. Wait 1 ms for DMA 1 to finish the transfer.
14. Read and verify that the Transfer Count Register is zero.
15. Read and verify the destination memory locations.

PROGRAMMABLE INTERRUPT CONTROLLER

16. Save current Interrupt Controller registers.
17. Write the pattern *0005hex* to the DMA 0 Control Register, DMA 1 Control Register, Timer Control Register, Interrupt 0 Control Register, Interrupt 1 Control Register, Interrupt 2 Control Register, Interrupt 3 Control Register, Priority Mask Register, and the Interrupt Status Register.
18. Read all the previous registers and verify that each contains the written pattern.
19. Repeat steps 17 and 18 using the pattern *000Ahex* for all registers except the last two. In those cases, use the pattern *0002hex*.
20. Set the priority for each internal interrupt.
 - Write *0003hex* to DMA 0 Control Register, *0004hex* to DMA 1 Control Register, *0005hex* to Timer Control Register, and *00FDhex* to the Mask Register.
21. Write *0000hex* to the Priority Mask Register.
22. Set up DMA 0 to generate an interrupt by writing *FF07hex* to DMA 0 Control Word Register for the first pass, DMA 1 to generate an interrupt by writing *FF07hex* to DMA 1 Control Word Register for the second pass, and Timer 2 to generate an interrupt by writing *C000hex* to Timer 2 Mode/Control Register for the third pass.
23. Read and verify that the Interrupt Request Register shows a pending interrupt for DMA 0.
24. Set the Mask Register to allow the interrupt by writing *00FBhex* to the Mask Register.
25. Read and verify that the In-Service Register and Poll Status Register show no DMA 0 interrupt.

26. Set the Priority Mask Register to `0003hex` to allow DMA 0 interrupt for the first pass, to `0004` for the second pass, and `0005hex` for third pass.
27. Read and verify that the Poll Status Register shows a DMA 0 interrupt.
28. Verify that the DMA 0 interrupt handler shows that the DMA 0 interrupt was serviced.
29. Read and verify that the In-Service Register and Interrupt Request Register show no DMA 0 interrupt.
30. Set the Mask Register to `00FDhex`.
31. Repeat steps 22 through 30, replacing DMA 0 with DMA 1 on the second pass and with Timer 2 on the third pass.

Error Index 6

The Timer 2 count was not within the expected tolerance; or the DMA Transfer Count Register was not zero; or one of the DMA transfers was not successful; or one of the interrupt registers did not contain the correct status. The test will endlessly repeat as a debugging aide.

Routine Name	MMU Control
Overview	This test verifies the MMU Mode Register U651 <19>. This is done by using a "walking ones" test.
Description	<ol style="list-style-type: none">1. Perform a "walking ones" test on the MMU Mode Register.<ul style="list-style-type: none">• Write <i>01hex</i> to the MMU Mode Register.• Read the MMU Mode Register and verify that the data read back is <i>01hex</i>.• Continue this write/read/verify sequence with the patterns <i>02hex</i>, <i>04hex</i>, <i>08hex</i>, <i>10hex</i>, <i>20hex</i>, <i>40hex</i>, <i>80hex</i>.
Error Index 7	The value read back did not match what was written. The test will endlessly repeat as a debugging aide.

Routine Name MMU Handshake

Overview This test verifies **TBC-MMU Handshake Control (Input/Output)** <19>, **TBC-MMU Next DMA Request FF (Flip-Flop)** <19>, **MMU Status Read Buffer** <19>, and partially checks **MMU Bidirectional Data Port** <19>. This is done by setting up the DMA channels to transfer a message and simulating the handshake signals from the Executive Processor.

Description

1. Clear any possible message requests by pulsing **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) and **TBC-MMU Next DMA Request FF** MMU_DRGLR(L), and reading the **MMU Bidirectional Data Port**.
2. Set up CPU 80186 DMA 0 <17> to receive a one word message and DMA 1 to send a one word message.
 - Load both DMA's Source Pointer and Destination Pointer Registers to point to a one word area of memory (actual location is not important), both DMA's Transfer Count Registers with 0001hex, and both DMA's Control Word Registers with 6E67hex.
3. Read **MMU Status Read Buffer** U851A to verify that **TBC-MMU Next DMA Request FF** U480A-5 DRQ1 is low and **MMU Bidirectional Data Port** U680-11 DRQ0 is low. Verify that DMA 0 Transfer Count Register is still 1, and DMA 1 Transfer Count Register is still 1.
4. Start a receive message operation by pulsing **TBC-MMU Handshake Control (Input)** MMU_AOUT(L) to simulate DIGLATCH(L) from the Executive Processor.
5. Read **MMU Status Read Buffer** to verify that the send DMA request (DRQ1) is low and that the receive DMA request (DRQ0) is high. Verify that DMA 1 Transfer Count Register is still 1 and that DMA 0 Transfer Count Register is now 0.
6. Clear the receive DMA request (DRQ0) by reading the **MMU Bidirectional Data Port**, through the **MMU Message Data Transceivers**.
7. Read **MMU Status Read Buffer** to verify that the send DMA request (DRQ1) is still low and that the receive DMA request (DRQ0) is now also low. Verify that DMA 1 Transfer Count Register is still 1 and that DMA 0 Transfer Count Register is still 0.
8. Start a send message operation by writing to the **MMU Bidirectional Data Port**, through the **MMU Message Data Transceivers**.
9. Read **MMU Status Read Buffer** to verify that there are no send or receive DMA requests present (i.e. DRQ1 and DRQ0 low). Verify that DMA 1 Transfer Count Register is still 1 and that DMA 0 Transfer Count Register is still 0.

10. Pulse **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) to simulate DOUT(L) from the Executive Processor.
11. Read **MMU Status Read Buffer** to verify that there are now send and receive DMA requests present (i.e. DRQ1 and DRQ0 high). Verify that DMA 1 Transfer Count Register is now 0 and that DMA 0 Transfer Count Register is still 0.
12. Clear the send DMA request (DRQ1) by pulsing **TBC-MMU Next DMA Request FF U491C-10 MMU_DRGLR(L)**.
13. Read **MMU Status Read Buffer** to verify that the send DMA request (DRQ1) is now low and that the receive DMA request (DRQ0) is still high. Verify that DMA 1 Transfer Count Register is still 0 and that DMA 0 Transfer Count Register is still 0.
14. Repeat the above steps 256 times.

Error Index 8

The state of the send or receive DMA request control signals (DRQ1, DRQ0), read from the **MMU Status Read Buffer** <19>, was not correct. The test will endlessly repeat as a debugging aide.

Routine Name MMU Messages

Overview This test verifies the MMU **Bidirectional Data Port** <19>, through the **MMU Message Data Transceivers** <19>. "Walking ones" patterns are written to the data port and read back by simulating the Executive Processor handshake.

Description

1. Write *0001hex* to the data port latches U583 & U680 <19>, through the data transceivers U561 & U661 <19>.
2. Pulse **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) <19> to strobe the test pattern back into the data port latches.
3. Read the pattern from the data port latches, through the data transceivers, and verify that it is *0001hex*.
4. Repeat above steps using the patterns *0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, 8000hex*.

Error Index 9 The pattern read back did not match the pattern written on the low or high order 8 bits. The test will endlessly repeat as a debugging aide.

Routine Name MMU Addressing

Overview This test verifies the **MMU Write Offset Address Register <19>** by using a "walking zeros" test and simulating the Executive Processor handshake.

Description

1. Write **FFFF_{hex}** to address **39FFE_{hex}**. This latches **FFFF_{hex}** into the **MMU Bidirectional Data Port <19>** (which is unimportant) and **1FFE_{hex}** into the **MMU Write Offset Address Register <19>**.
2. Pulse **MMU_AOUT(L)** to latch the address from the **MMU Write Offset Address Register** into the **MMU Bidirectional Data Port**.
3. Read the address pattern from the **MMU Bidirectional Data Port**, through the **MMU Message Data Transceivers**, and verify that it is **1FFE_{hex}**.
4. Repeat above steps writing **FFFF_{hex}** to the following addresses and verifying the associated expected pattern from the data port latches.

<u>Address</u>	<u>Expected Pattern</u>
39FFD _{hex}	1FFD _{hex}
39FFB _{hex}	1FFB _{hex}
39FF7 _{hex}	1FF7 _{hex}
39FEF _{hex}	1FEF _{hex}
39FDF _{hex}	1FDF _{hex}
39FBF _{hex}	1FBF _{hex}
39F7F _{hex}	1F7F _{hex}
39EFF _{hex}	1EFF _{hex}
39DFF _{hex}	1DFF _{hex}
39BFF _{hex}	1BFF _{hex}
397FF _{hex}	17FF _{hex}
38FFF _{hex}	0FFF _{hex}

Error Index 9 One of the patterns read back from the low or high bytes did not match what was expected. The test will endlessly repeat as a debugging aide.

Routine Name Acquisition Configuration

Overview This test attempts to communicate with all possible acquisition systems. The result of this test determines what acquisition systems are available for acquiring data. The test exercises TBC-ACQ Data Transceivers <20>, TBC-ACQ Address Buffers <20>, and TBC-ACQ Control Logic <20>.

Description 1. Perform address uniqueness test on the shared communication buffer memory.

- Write a 0001_{hex} to the M/F acquisition system 1 memory location which is at an offset of 2BA8_{hex} above the memory address given in the table below (i.e. 22BA8_{hex}). Now write a 0002_{hex} to the M/F acquisition system 2 memory location which is at an offset of 2BA8_{hex} above the memory address given in the table below (i.e. 26BA8_{hex}). Continue this procedure, using the patterns 0003_{hex}, 0004_{hex},..., 0021_{hex}, 0022_{hex} to each successive acquisition system.
- Read from each acquisition system and verify that the value written in the previous step is still present. Any acquisition system with an incorrect value is removed from the list of active acquisition systems and the next four steps are skipped.
- Fill the memory address range, from offset 2BA4_{hex} to offset 2BF4_{hex} above the memory address given in the table below (i.e. 22BA4_{hex} to 22BF4_{hex}), with the pattern AAAA_{hex}.
- Read and verify address offset 2BA4_{hex} for AAAA_{hex}. If so, write CCCC_{hex} to address offset 2BA4_{hex}. Increment the address and continue this read/verify/write sequence until address offset 2BF4_{hex} is reached.
- Repeat the read/verify/write sequence, starting again at address offset 2BA4_{hex}, for F0F0_{hex}, 5555_{hex}, and AAAA_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern).
- Make one last read/verify pass for the pattern AAAA_{hex}, starting at address offset 2BA4_{hex}.
- Write a FFFF_{hex} to the M/F acquisition system 1 memory location which is at an offset of 2BA8_{hex} above the memory address given in the table below (i.e. 22BA8_{hex}). Now write a FFFE_{hex} to the M/F acquisition system 2 memory location which is at an offset of 2BA8_{hex} above the memory address given in the table below (i.e. 26BA8_{hex}). Continue this procedure, using the patterns FFFD_{hex}, FFFC_{hex},..., FFDF_{hex}, FFDE_{hex} to each successive acquisition system.
- Read from each acquisition system and verify that the value written in the previous step is still present. Any acquisition system

with an incorrect value is removed from the list of active acquisition systems.

2. Clear any pending interrupts from all acquisition systems below.
 - Read from the interrupt addresses, plus address offset 2, given in the table.
3. Load the test patterns into each acquisition system that passed step 1.
 - Transfer 0001hex, 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, 8000hex, FFFEhex, FFFDhex, FFFBhex, FFF7hex, FFEFhex, FFDFhex, FFBFhex, FF7Fhex, FEFFhex, FDFhex, FBFFhex, F7FFhex, EFFFhex, DFFFhex, BFFFhex, 7FFFhex, 0001hex, 0002hex, 0004hex to the memory addresses, plus offset 2BA4hex, given in the table below.
4. Send an interrupt to each acquisition system that passed step 1.
 - Write to the interrupt address given below.
5. Read back the test patterns that were written earlier and verify that they have been inverted by the acquisition system.
 - Test patterns should be FFFEhex, FFFDhex, FFFBhex, FFF7hex, FFEFhex, FFDFhex, FFBFhex, FF7Fhex, FEFFhex, FDFhex, FBFFhex, F7FFhex, EFFFhex, DFFFhex, BFFFhex, 7FFFhex, 0001hex, 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, 8000hex, FFFEhex, FFFDhex, FFFBhex.
6. Read and verify that the acquisition system returned an interrupt to the timebase.
 - Read the interrupt address given below and verify that the status bit given below is set.

Error Index B

No errors are possible. This error code is only used to show the sequence of events, as displayed on the LEDs, during power-up diagnostics.

Timebase Kernel

Acquisition Configuration (B)

Address Table for 11801

	M/F	MCU A	MCU B	MCU C	MCU D
Acq 1					
Memory	20000	40000	60000	80000	A0000
Interrupt	30200	30240	30260	30280	302A0
Status	bit 0	bit 0	bit 8	bit 0	bit 8
Acq 2					
Memory	24000	44000	64000	84000	A4000
Interrupt	30204	30244	30264	30284	302A4
Status	bit 1	bit 1	bit 9	bit 1	bit 9
Acq 3					
Memory		48000	68000	88000	A8000
Interrupt		30248	30268	30288	302A8
Status		bit 2	bit 10	bit 2	bit 10
Acq 4					
Memory		4C000	6C000	8C000	AC000
Interrupt		3024C	3026C	3028C	302AC
Status		bit 3	bit 11	bit 3	bit 11
Acq 5					
Memory		50000	70000	90000	B0000
Interrupt		30250	30270	30290	302B0
Status		bit 4	bit 12	bit 4	bit 12
Acq 6					
Memory		54000	74000	94000	B4000
Interrupt		30254	30274	30294	302B4
Status		bit 5	bit 13	bit 5	bit 13
Acq 7					
Memory		58000	78000	98000	B8000
Interrupt		30258	30278	30298	302B8
Status		bit 6	bit 14	bit 6	bit 14
Acq 8					
Memory		5C000	7C000	9C000	BC000
Interrupt		3025C	3027C	3029C	302BC
Status		bit 7	bit 15	bit 7	bit 15

Address Table for 11802

Acq 1	
Memory	24000
Interrupt	30204
Status	bit 1

Routine Name Executive Communication

Overview This test verifies the communication path from the Timebase processor to the Executive processor via **TBC-MMU Handshake Control (Input/Output) <19>**, **TBC-MMU Next DMA Request FF (Flip-Flop) <19>**, **MMU Status Read Buffer <19>**, **MMU Bidirectional Data Port <19>**, and **MMU Write Offset Address Register <19>** by inverting and echoing "walking one's" and "walking zero's" patterns which are expected to be received from the Executive processor. If, at any time, the Timebase processor receives an unexpected pattern, it enters a "forever loop" in which it inverts and echos every pattern sent thereafter.

Description

1. Enable **TBC-MMU Handshake Control (Input/Output) <19>** to allow data patterns to be sent from the Timebase to the Executive.
2. Set up **Programmable DMA Unit's U440 <17>** channel 0 to receive a data pattern from the **MMU Bidirectional Data Port <19>** without generating any interrupt at the end of the transfer.
3. Set up **Programmable DMA Unit's U440 <17>** channel 1 to send a data pattern from **General Purpose Static RAM <17>** to the **MMU Bidirectional Data Port <19>** without generating any interrupt at the end of the transfer.
4. Wait for a data pattern to be received from the Executive processor through DMA 0.
5. If the received data pattern matches the expected pattern, then invert all bits (i.e., 1's complement) and allow DMA 1 to send the inverted pattern back to the Executive processor (see Error Index section below).
6. Signal the Executive processor, via the MMU, that a data pattern has been sent by a writing *0Fhex* to **MMU Mode Register <19>** and then writing a *0000hex* to **MMU Bidirectional Data Port <19>**.
7. If the received data pattern did not match the expected pattern, perform an infinite loop in which any patterns received, as in step 4, are inverted and sent back to the Executive processor, as in steps 5 & 6.
8. Repeat steps 4-6 until all expected patterns have been received, inverted, and sent back (see Error Index section below).

Error Index C One of the data patterns received from the Executive processor did not match the expected pattern. The expected patterns to be received and those returned to the Executive processor are shown in the following table:

Timebase Kernel

Executive Communication (C)

Expected Pattern Received (<i>hex</i>)	Pattern Sent (<i>hex</i>)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	FFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
FFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001

Routine Name EPROM U611 Location/Checksum

Overview This test verifies the location and integrity of EPROM U611 <12>, MPU & Bus Buffers U410, U500, U510 <12>, and partially verifies Address Decoder U610 <12> by performing a complement, location, and checksum test on EPROM U611.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Set EPROM U611-1 BANKSEL <12> low.
 - Exclusive-or the contents of byte locations F004_{hex} and F005_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.
3. If the location test was successful (step 2), then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U611 except the first two, and then verify it against the checksum stored in the first two bytes of the device.

Error Index 1 The bytes at F004_{hex} and F005_{hex} were not complementary; or the location byte at F006_{hex} did not match the expected value; or the checksum for the device was not correct. After displaying the error code on the LEDs, the processor goes into an endless loop branching to the same location.

Routine Name RAM Data/Address

Overview This test verifies the data and address lines from MPU U401 <12>, through MPU & Bus Buffers <12>, to the Scratchpad RAM <12> and Shared RAM <12> by performing a "walking one's" test on three memory locations and performing a RAM test on both sections of memory. Address Decoder U610 <12> and Bus Arbiter <13> are also partially verified.

Description

1. Initialize the test locations.
 - Write the pattern FF_{hex} to address 0000_{hex} , 0001_{hex}, and 4000_{hex}.
2. Verify Address Decoder U610 <12>.
 - Read address 0000_{hex} and verify that it is FF_{hex}.
3. Perform a "walking one's" test on Shared RAM <12> address 0000_{hex}. Terminate test if any verify operation fails.
 - Write the pattern 80_{hex} to address 0000_{hex}. Read the same address and verify that it was 80_{hex}. Continue this write/read/verify sequence with the patterns 40_{hex}, 20_{hex}, 10_{hex}, 08_{hex}, 04_{hex}, 02_{hex}, 01_{hex}.
4. Verify Address Decoder U610 <12>.
 - Write 00_{hex} to address 0000_{hex}.
 - Read address 0001_{hex} and verify that it is FF_{hex}.
5. Perform a "walking one's" test on Shared RAM <12> address 0001_{hex}. Terminate test if any verify operation fails.
 - Write the pattern 80_{hex} to address 0001_{hex}. Read the same address and verify that it was 80_{hex}. Continue this write/read/verify sequence with the patterns 40_{hex}, 20_{hex}, 10_{hex}, 08_{hex}, 04_{hex}, 02_{hex}, 01_{hex}.
6. Verify Address Decoder U610 <12>.
 - Write 00_{hex} to address 0001_{hex}.
 - Read address 4000_{hex} and verify that it is FF_{hex}.
7. Perform a "walking one's" test on Scratchpad RAM <12> address 4000_{hex}. Terminate test if any verify operation fails.
 - Write the pattern 80_{hex} to address 4000_{hex}. Read the same address and verify that it was 80_{hex}. Continue this write/read/verify sequence with the patterns 40_{hex}, 20_{hex}, 10_{hex}, 08_{hex}, 04_{hex}, 02_{hex}, 01_{hex}.

8. Verify **Address Decoder U610 <12>**.
 - Write *00hex* to address *4000hex*.
 - Read address *0000hex* , *0001hex*, and *4000hex* and verify that they are *00hex*.
9. Verify **Shared RAM <12>** address range *2A08hex* to *2BA2hex*.
Terminate test if any verify operation fails.
 - Fill address range *2A08hex* to *2BA2hex* with the pattern *AAhex*.
 - Read and verify address *2A08hex* for *AAhex*. If so, write *55hex* to address *2A08hex*. Increment the address and continue this read/verify/write sequence until address *2BA2hex* is reached.
 - Repeat the read/verify/write sequence, starting again at address *2A08hex*, for *FFhex* and *00hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
 - Make one last read/verify pass for the pattern *00hex* starting at address *2A08hex*.
10. Verify **Shared RAM and Scratchpad RAM<12>** address range *2C48hex* to *5FFFhex*. Terminate test if any verify operation fails.
 - Fill address range *2C48hex* to *5FFFhex* with the pattern *AAhex*.
 - Read and verify address *2C48hex* for *AAhex*. If so, write *55hex* to address *2C48hex*. Increment the address and continue this read/verify/write sequence until address *5FFFhex* is reached.
 - Repeat the read/verify/write sequence, starting again at address *2C48hex*, for *FFhex* and *00hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
 - Make one last read/verify pass for the pattern *00hex* starting at address *2C48hex*.

Error Index 2

One of the patterns read from a memory location in the **Shared RAM** or **Scratchpad RAM <12>** was not the pattern written. The test will endlessly write and read the failing test pattern to the failing memory location as a debugging aide.

Routine Name Timebase Communication

Overview This test verifies the communication path from the Acquisition processor to the Timebase processor via **TBC Interface Decoding <14>**, **TBC Interface Transceivers <14>**, **TBC Interrupt Latch <14>**, **Bus Arbiter <13>**, and **Shared RAM <12>** by inverting and echoing "walking one's" and "walking zero's" patterns which are expected to be received from the Timebase processor. If, at any time, the Acquisition processor receives an unexpected pattern, it enters a "forever loop" in which it inverts and echos every pattern sent thereafter.

Description

1. Enable IRQ interrupts.
2. Wait for interrupt from Timebase processor.
3. Disable IRQ interrupts.
4. Check received data patterns (see table below).
 - If a received data pattern does not match the expected pattern, then flag the data pattern as an error.
 - Invert all bits (i.e., 1's complement) in the test patterns and write the test patterns back to the same place they were read from.
5. Signal the Timebase processor by setting **TBC Interrupt Latch U210-5 INTR <14>** high.
6. If one or more received data patterns did not match the expected pattern, perform an infinite loop in which any patterns received are inverted and sent back to the Timebase processor, as in steps 4 & 5.

Error Index 3 One or more of the data patterns received from the Timebase processor did not match the expected pattern. The expected patterns to be received and those returned to the Timebase processor are shown in the following table:

Acquisition Kernels

Timebase Communication (3)

Expected Pattern Received (<i>hex</i>)	Pattern Sent (<i>hex</i>)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	EFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
EFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001
0001	FFFE
0002	FFFD
0004	FFFB

Routine Name U250

Overview This test verifies that ROM U250 <28> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations E0008_{hex} and E000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1111 The bytes at E0008_{hex} and E000A_{hex} were not complementary.

Error Index E1112 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U630

Overview This test verifies that EPROM U630 <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations C0008_{hex} and C000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1121 The bytes at C0008_{hex} and C000A_{hex} were not complementary.

Error Index E1122 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U600

Overview This test verifies that EPROM U600 <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations C0008_{hex} and C000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1131 The bytes at C0008_{hex} and C000A_{hex} were not complementary.

Error Index E1132 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U612

Overview This test verifies that EPROM U612 <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations A0008_{hex} and A000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1141 The bytes at A0008_{hex} and A000A_{hex} were not complementary.

Error Index E1142 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U620

Overview This test verifies that EPROM U620 <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations 80008_{hex} and 8000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1151 The bytes at 80008_{hex} and 8000A_{hex} were not complementary.

Error Index E1152 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Memory

ROM Loc Lo

U210 Mem Xpn (E116X)

Routine Name U210 Mem Xpn (Memory Expansion)

Overview This test verifies that RAM/ROM U210 <32> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the RAM/ROM <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations A0008_{hex} and A000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the RAM/ROM <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1161 The bytes at A0008_{hex} and A000A_{hex} were not complementary.

Error Index E1162 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Executive

Routine Name U240 Mem Xpn (Memory Expansion)

Overview This test verifies that **RAM/ROM U240 <32>** is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Exclusive-or the contents of byte locations *80008hex* and *8000Ahex* and verify that the result is *FFhex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.

Error Index E1171 The bytes at *80008hex* and *8000Ahex* were not complementary.

Error Index E1172 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U240

Overview This test verifies that EPROM U240 <28> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations E0009_{hex} and E000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1211 The bytes at E0009_{hex} and E000B_{hex} were not complementary.

Error Index E1212 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U730

Overview This test verifies that EPROM U730 <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations C0009_{hex} and C000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1221 The bytes at C0009_{hex} and C000B_{hex} were not complementary.

Error Index E1222 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U700

Overview This test verifies that **EPROM U700** <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select U530A** <27>.
 - Select the **EPROM bank 1** by setting **Bank Decode/Select U520-6** <27> high and **U520-11** <27> low.
 - Exclusive-or the contents of byte locations **C0009_{hex}** and **C000B_{hex}** and verify that the result is **FF_{hex}**.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520** <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select U530A** <27>.
 - Select the **EPROM bank 1** by setting **Bank Decode/Select U520-6** <27> high and **U520-11** <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520** <27>.

Error Index E1231 The bytes at **C0009_{hex}** and **C000B_{hex}** were not complementary.

Error Index E1232 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U712

Overview This test verifies that EPROM U712 <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations A0009_{hex} and A000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1241 The bytes at A0009_{hex} and A000B_{hex} were not complementary.

Error Index E1242 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U720

Overview This test verifies that EPROM U720 <29> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations 80009_{hex} and 8000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1251 The bytes at 80009_{hex} and 8000B_{hex} were not complementary.

Error Index E1252 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name U220 Mem Xpn (Memory Expansion)

Overview This test verifies that RAM/ROM U220 <32> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **RAM/ROM** <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations A0009_{hex} and A000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **RAM/ROM** <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1261 The bytes at A0009_{hex} and A000B_{hex} were not complementary.

Error Index E1262 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Memory

ROM Loc Hi

U230 Mem Xpn (E127X)

Routine Name U230 Mem Xpn (Memory Expansion)

Overview This test verifies that **RAM/ROM U230 <32>** is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Exclusive-or the contents of byte locations **80009_{hex}** and **8000B_{hex}** and verify that the result is **FF_{hex}**.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select U530A <27>**.
 - Select the **RAM/ROM <32>** bank 0 by setting **Bank Decode/Select U520-6,11 <27>** low.
 - Read and verify the location byte against the known value.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select U520 <27>**.

Error Index E1271 The bytes at **80009_{hex}** and **8000B_{hex}** were not complementary.

Error Index E1272 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Executive

Routine Name U250

Overview This test verifies the integrity of ROM U250 <28> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U250 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1311 The bytes at E0008*hex* and E000A*hex* were not complementary.

Error Index E1312 The computed checksum did not match the stored checksum.

Routine Name U630

Overview This test verifies the integrity of EPROM U630 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U630 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1321 The bytes at C0008*hex* and C000A*hex* were not complementary.

Error Index E1322 The computed checksum did not match the stored checksum.

Routine Name U600

Overview This test verifies the integrity of EPROM U600 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations C0008_{hex} and C000A_{hex} and verify that the result is FF_{hex}
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Perform a checksum on all the bytes in U600 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1331 The bytes at C0008_{hex} and C000A_{hex} were not complementary.

Error Index E1332 The computed checksum did not match the stored checksum.

Routine Name U612

Overview This test verifies the integrity of EPROM U612 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations A0008_{hex} and A000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Perform a checksum on all the bytes in U612 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1341 The bytes at A0008_{hex} and A000A_{hex} were not complementary.

Error Index E1342 The computed checksum did not match the stored checksum.

Routine Name U620

Overview This test verifies the integrity of EPROM U620 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations 80008*hex* and 8000A*hex* and verify that the result is FF*hex*.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Perform a checksum on all the bytes in U620 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1351 The bytes at 80008*hex* and 8000A*hex* were not complementary.

Error Index E1352 The computed checksum did not match the stored checksum.

Routine Name U210 Mem Xpn (Memory Expansion)

Overview This test verifies the integrity of RAM/ROM U210 <32> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **RAM/ROM** <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations A0008_{hex} and A000A_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **RAM/ROM** <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U210 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1361 The bytes at A0008_{hex} and A000A_{hex} were not complementary.

Error Index E1362 The computed checksum did not match the stored checksum.

Routine Name	U240 Mem Xpn (Memory Expansion)
Overview	This test verifies the integrity of RAM/ROM U240 <32> by performing a checksum on its contents.
Description	<ol style="list-style-type: none">1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.<ul style="list-style-type: none">• Store the current bank selection by reading Bank Decode/Select U530A <27>.• Select the RAM/ROM <32> bank 0 by setting Bank Decode/Select U520-6,11 <27> low.• Exclusive-or the contents of byte locations 80008_{hex} and 8000A_{hex} and verify that the result is FF_{hex}.• Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.<ul style="list-style-type: none">• Store the current bank selection by reading Bank Decode/Select U530A <27>.• Select the RAM/ROM <32> bank 0 by setting Bank Decode/Select U520-6,11 <27> low.• Perform a checksum on all the bytes in U230 except the first two and verify it against the checksum stored in the first two bytes of the device.• Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.
Error Index E1371	The bytes at 80008 _{hex} and 8000A _{hex} were not complementary.
Error Index E1372	The computed checksum did not match the stored checksum.

Routine Name U240

Overview This test verifies the integrity of ROM U240 <28> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations E0009_{hex} and E000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U240 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1411 The bytes at E0009_{hex} and E000B_{hex} were not complementary.

Error Index E1412 The computed checksum did not match the stored checksum.

Routine Name U730

Overview This test verifies the integrity of EPROM U730 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations C0009_{hex} and C000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U730 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1421 The bytes at C0009_{hex} and C000B_{hex} were not complementary.

Error Index E1422 The computed checksum did not match the stored checksum.

Routine Name U700

Overview This test verifies the integrity of EPROM U700 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations C0009_{hex} and C000B_{hex} and verify that the result is FF_{hex}
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **EPROM** bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Perform a checksum on all the bytes in U700 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1431 The bytes at C0009_{hex} and C000B_{hex} were not complementary.

Error Index E1432 The computed checksum did not match the stored checksum.

Routine Name U712

Overview This test verifies the integrity of EPROM U712 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations A0009_{hex} and A000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Perform a checksum on all the bytes in U712 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1441 The bytes at A0009_{hex} and A000B_{hex} were not complementary.

Error Index E1442 The computed checksum did not match the stored checksum.

Routine Name U720

Overview This test verifies the integrity of EPROM U720 <29> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Exclusive-or the contents of byte locations 80009_{hex} and 8000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the EPROM bank 1 by setting **Bank Decode/Select** U520-6 <27> high and U520-11 <27> low.
 - Perform a checksum on all the bytes in U720 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1451 The bytes at 80009_{hex} and 8000B_{hex} were not complementary.

Error Index E1452 The computed checksum did not match the stored checksum.

Routine Name	U220 Mem Xpn (Memory Expansion)
Overview	This test verifies the integrity of RAM/ROM U220 <32> by performing a checksum on its contents.
Description	<ol style="list-style-type: none">1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.<ul style="list-style-type: none">• Store the current bank selection by reading Bank Decode/Select U530A <27>.• Select the RAM/ROM <32> bank 0 by setting Bank Decode/Select U520-6,11 <27> low.• Exclusive-or the contents of byte locations A0009_{hex} and A000B_{hex} and verify that the result is FF_{hex}.• Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.<ul style="list-style-type: none">• Store the current bank selection by reading Bank Decode/Select U530A <27>.• Select the RAM/ROM <32> bank 0 by setting Bank Decode/Select U520-6,11 <27> low.• Perform a checksum on all the bytes in U220 except the first two and verify it against the checksum stored in the first two bytes of the device.• Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.
Error Index E1461	The bytes at A0009 _{hex} and A000B _{hex} were not complementary.
Error Index E1462	The computed checksum did not match the stored checksum.

Memory

ROM Cksum Hi

U230 Mem Xpn (E147X)

Routine Name U230 Mem Xpn (Memory Expansion)

Overview This test verifies the integrity of RAM/ROM U230 <32> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **RAM/ROM** <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Exclusive-or the contents of byte locations 80009_{hex} and 8000B_{hex} and verify that the result is FF_{hex}.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
 - Select the **RAM/ROM** <32> bank 0 by setting **Bank Decode/Select** U520-6,11 <27> low.
 - Perform a checksum on all the bytes in U240 except the first two and verify it against the checksum stored in the first two bytes of the device.
 - Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1471 The bytes at 80009_{hex} and 8000B_{hex} were not complementary.

Error Index E1472 The computed checksum did not match the stored checksum.

Executive

Memory	RAM Refresh	Rate (E151X)
---------------	--------------------	---------------------

Routine Name	Rate
---------------------	-------------

Overview	This test verifies the refresh rate of DRAM Controller U410 <30> . A combination of refresh RAS signals is routed to the Timer U822 COUNTER 2 <25> through DIAGNSIG(L) <29> and is counted for a short duration.
-----------------	---

Description	<ol style="list-style-type: none"> 1. Inhibit the counter from counting. <ul style="list-style-type: none"> • Write <i>40hex</i> to Timer Configuration Logic U720 <25> so that G2 U822-16:L <25>. 2. Program the counter to count down with <i>6464hex</i> as the starting counter value. <ul style="list-style-type: none"> • Write <i>B0hex</i>, <i>64hex</i>, and <i>64hex</i> to Timer Control Word Register and Count Register low and high bytes, respectively. 3. Route Memory Diagnostic Signal Select's <29> combined ROW COLUMN BUS <29> signal to DIAGNSIG(L) <29> on the Executive Bus by writing <i>02hex</i> to U322B <29>. 4. Enable the counter to count DIAGNSIG(L) <25>. <ul style="list-style-type: none"> • Write <i>80hex</i> to U720 <25>. This sets G2 U822-16:H and connects DIAGNSIG(L) <25> to CLK2 U822-18. 5. Perform a software delay loop. 6. Disable the counter from counting. <ul style="list-style-type: none"> • Write <i>40hex</i> to Timer Configuration Logic U720 <25> so that the gate signal G2 U822-16 is low. 7. Read and verify Timer COUNTER 2 Count Register low and high bytes. 8. Stop Timer COUNTER 2 by programming its Control Word Register with <i>B0hex</i>. 9. Disable Memory Diagnostic Signal Select's <29> combined ROW COLUMN BUS <29> signal so that it does not appear in the Executive Bus by writing <i>00hex</i> to U352B <29>.
--------------------	---

Error Index E1511	The refresh rate was not within the expected tolerance.
--------------------------	---

Caveats	Other circuits that gate signals on to bus signal DIAGNSIG(L) may cause this test to fail. This test may also fail if the Main Processor Board Clock Generator <27> reference Y950 is out of tolerance since it is being used as the timing reference for the test via software delay mechanism.
----------------	--

Executive

Routine Name	Config (Configuration)
Overview	This test verifies DRAM Configuration <30> jumper selections and the software bank selectability through Bank Decode/Select <27> circuit.
Description	<ol style="list-style-type: none"> 1. Read the Memory Configuration Readback U832 <29>. Only the upper four bits of this port represent the jumper positions. The lower four bits represent the current bank selection. 2. Verify the position of the EPROM size selection jumper J541 <29> for 27512 type EPROMs. 3. Verify the position of the virtual or real memory addressing jumper J520 <30> for real memory addressing. 4. Verify the combination of RAM device type and number of RAM banks jumper J501 <30> and J521 <30>, respectively, for 64k X 4 RAM device type and two banks. 5. Store the current bank by reading Bank Decode/Select U530A <27>. 6. Verify bank selectability by doing a "walking one's" test on the 4-bit bank select register U520 <27>. <ul style="list-style-type: none"> • Write patterns 01hex, 02hex, 04hex, and 08hex to Bank Decode/Select U520 <27>. Read and verify the written patterns from Bank Decode/Select U530A <27> and Memory Configuration Readback U832 <29> (only the lower four bits). 7. Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.
Error Index E1611	27512 EPROM has not been selected through J541.
Error Index E1612	Real mode selection has not been made through J520.
Error Index E1613	The RAM device and number of banks combination selected through J501 and J521, respectively, are not appropriate.
Error Index E1614	The bank select mechanism is not functioning properly. The display shows the first pattern that did not verify.

Memory

Dynamic RAM

Data Lines (E162X)

Routine Name

Data Lines

Overview

This test verifies the the data lines to DRAM <30> by performing a "walking one's" test on DRAM address 3BFFE_{hex}.

Description

1. Store the current bank selection by reading Bank Decode/Select U530A <27>.
2. Select DRAM bank 1 by setting Bank Decode/Select U520-3:H <27>.
3. Perform a "walking one's" test on DRAM address 3BFFE_{hex}.
 - Write the pattern 0001_{hex} to address 3BFFE_{hex}. Read the same address and verify that it was 0001_{hex}. Continue this write/read/verify sequence with the patterns 0002_{hex}, 0004_{hex}, 0008_{hex}, 0010_{hex}, 0020_{hex}, 0040_{hex}, 0080_{hex}, 0100_{hex}, 0200_{hex}, 0400_{hex}, 0800_{hex}, 1000_{hex}, 2000_{hex}, 4000_{hex}, and 8000_{hex}.
4. Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.

Error Index E1621

The pattern read from DRAM <30> address 3BFFE_{hex} was not the same pattern written. The display shows the first pattern that did not verify. The following table shows the RAM device that is probably bad.

Data Bits	Device
0-3	U222
4-7	U220
8-11	U212
12-15	U210

Memory	Dynamic RAM	Address/Data (E163X)
Routine Name	Address/Data	
Overview	This test verifies the address lines and data integrity of the DRAM <30> by performing a RAM test on all DRAM locations.	
Description	<ol style="list-style-type: none"> 1. Store the current bank selection by reading Bank Decode/Select U530A <27>. 2. Select DRAM memory bank 1 by setting Bank Decode/Select U520-3:H <27>. 3. Verify DRAM address range 2C000-3BFFF_{hex} (64k bytes). Terminate test if any verify operation fails. <ul style="list-style-type: none"> • Fill address range 2C000-3BFFF_{hex} with the pattern AAAA_{hex}. • Read and verify address 2C000_{hex} for AAAA_{hex}. If so, write CCCC_{hex} to address 2C000_{hex}. Increment address and continue this read/verify/write sequence until address 3BFFF_{hex} is reached. • Repeat the read/verify/write sequence, starting again at address 2C000_{hex} for CCCC_{hex}, 5555_{hex}, and 0000_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern). 4. If DRAM address range 2C000-3BFFF_{hex} is free of faults, then test DRAM address ranges 00000-0FFFF_{hex} (64k bytes), 10000-1FFFF_{hex} (64k bytes), and 20000-2BFFF_{hex} (48k bytes) separately. <ul style="list-style-type: none"> • Copy the 64k bytes or 48k bytes of DRAM to the uppermost 64k bytes (2C000-3BFFF_{hex}). • Test DRAM locations using the procedure outlined in step 3. • Restore the DRAM locations that were tested by copying back information stored in uppermost 64k bytes. 5. Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>. 	
Error Index E1631	The pattern read from the displayed memory location was not the pattern written. The following table may be used to help isolate the bad RAM device.	

Executive

Memory

Dynamic RAM

Address/Data (E163X)

Address	Data Bits			
	0 – 3	4 – 7	8 – 11	11 – 15
XXXX0 _{hex}	U122	U120	U112	U110
XXXX1 _{hex}	U122	U120	U112	U110
XXXX4 _{hex}	U122	U120	U112	U110
XXXX5 _{hex}	U122	U120	U112	U110
XXXX8 _{hex}	U122	U120	U112	U110
XXXX9 _{hex}	U122	U120	U112	U110
XXXXC _{hex}	U122	U120	U112	U110
XXXXD _{hex}	U122	U120	U112	U110
XXXX2 _{hex}	U222	U220	U212	U210
XXXX3 _{hex}	U222	U220	U212	U210
XXXX6 _{hex}	U222	U220	U212	U210
XXXX7 _{hex}	U222	U220	U212	U210
XXXXA _{hex}	U222	U220	U212	U210
XXXXB _{hex}	U222	U220	U212	U210
XXXXE _{hex}	U222	U220	U212	U210
XXXXF _{hex}	U222	U220	U212	U210

X—don't care.

Memory

NVRAM NoBank

Battery (E171X)

Routine Name Battery

Overview This test verifies the **Non-Volatile RAM** <28> volatility (i.e., that **RAM Battery** <28> was functional on power-up) by checking the values of two confidence words located in the NVRAM.

Description

1. Read and verify that the confidence word at NVRAM location `3C000hex` is `DEADhex`.
2. Read and verify that the confidence word at NVRAM location `3C002hex` is `2152hex` (complement of `DEADhex`).

Error Index E1711 The confidence word at `3C000hex` was not correct.

Error Index E1712 The confidence word at `3C002hex` was not correct.

Caveats The confidence words are initialized by the normal operating firmware in the system. So, if the instrument has never entered normal operating mode, this test will fail. Once initialized, the confidence words should remain complementary as long as the **RAM Battery** <28> is good.

Routine Name Data Lines

Overview This test verifies the data lines to the **Non-Volatile RAM** <28> by performing a "walking one's" test on NVRAM location 3C000hex.

CAUTION

*Turning the instrument power off during the execution of this test may result in losing some of the **Non-Volatile RAM** data used in normal operation and cause an NVRAM battery test failure to occur. Operation should not be seriously affected, however.*

Description

1. Perform a "walking one's" test on **Non-Volatile RAM** address 3C000hex.
 - Write the pattern 0001hex to **Non-Volatile RAM** address 3C000hex. Read the same address and verify that it was 0001hex. Continue this write/read/verify sequence with patterns 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, and 8000hex.

Error Index E1721 The pattern read did not match the pattern written. The correspondence of the data bits to RAM devices is shown in the following table:

Data Bits	Device
0 – 7	U270
8 – 15	U260

Caveats Turning the instrument power off during the execution of this test may result in losing the **Non-Volatile RAM** contents of the location being tested. This will cause the normal operating firmware to reset all of the **Non-Volatile RAM** contents and cause the previous diagnostic NVRAM battery test to fail.

Memory

NVRAM NoBank

Address/Data (E173X)

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of the Non-Volatile RAM <28> by performing a RAM test on all NVRAM locations.

CAUTION

Turning the instrument power off during the execution of this test may result in losing some of the Non-Volatile RAM data used in normal operation and cause an NVRAM battery test failure to occur. Operation should not be seriously affected, however.

Description

1. Verify Odd DRAM <34> and Even DRAM <34> address range 40000-43FFF_{hex} (16k bytes). Terminate test if any verify operation fails.
 - Fill address range 40000-43FFF_{hex} with the pattern AAAA_{hex}.
 - Read and verify address 40000_{hex} for AAAA_{hex}. If so, write CCCC_{hex} to address 40000_{hex}. Increment the address and continue this write/read/verify sequence until address 43FFF_{hex} is reached.
 - Repeat the read/verify/write sequence, starting again at address 40000_{hex}, for CCCC_{hex}, 5555_{hex}, and 0000_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern).
2. If Odd DRAM <34> and Even DRAM <34> address range 40000-43FFF_{hex} is free of faults, then verify Non-Volatile RAM <28> address range 3C000-3FFFF_{hex}.
 - Save data in Non-Volatile RAM address range 3C000-3FFFF_{hex} (16k bytes) by copying it to Odd DRAM and Even DRAM address range 40000-43FFF_{hex}.
 - Perform the same procedure as in step 1 for address range 3C000-3FFFF_{hex}, using patterns AAAA_{hex}, CCCC_{hex}, F0F0_{hex}, 5555_{hex}, and AAAA_{hex}.
 - Make one last read/verify pass for the pattern AAAA_{hex} starting at address 3C000_{hex}.
 - Copy Non-Volatile RAM data from Odd DRAM and Even DRAM address range 38000-3BFFF_{hex} to Non-Volatile RAM address range 3C000-3FFFF_{hex}. In order to restore the Non-Volatile RAM confidence words last, the copying is done from the last address to the first (i.e., from 3FFFE_{hex} to 3C000_{hex}).

Error Index E1731

The pattern read from Odd DRAM <34> and Even DRAM <34>, or the pattern read from Non-Volatile RAM <28>, was not the pattern written.

Executive

Memory

NVRAM NoBank

Address/Data (E173X)

If the displayed memory location is between 3C000-3FFFF_{hex}, then the correspondence of the data bits to **Non-Volatile RAM** <28> devices is shown in the following table:

Data Bits	Device
0–7	U270
8–15	U260

See Also

Executive Subsys Comm Waveform RAM Address/Data (E523X) if the faulty address was between 40000-43FFF_{hex}.

Caveats

Turning the instrument power off during the execution of this test may result in losing the **Non-Volatile RAM** contents of some or all of the locations being tested. This will cause the normal operating firmware to reset all of the **Non-Volatile RAM** contents and cause the previous diagnostic NVRAM battery test to fail.

Executive

Memory

NVRAM Banks

Battery (E181X)

Routine Name Battery

Overview This test verifies the **RAM/ROM** <32> volatility (i.e., that **Battery Backup** <31> batteries BT160 & BT260 were functional on power-up) by checking the values of two confidence words located in battery backed-up RAM (U410/U420).

Description

1. Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
2. Select **RAM/ROM** <32> bank 0 by setting **Bank Decode/Select** U520-2 & 5 <27> low.
3. Read and verify that the confidence word at **RAM/ROM** <32> location 20000_{hex} is DEAD_{hex}.
4. Read and verify that the confidence word at **RAM/ROM** <32> location 20002_{hex} is 2152_{hex} (complement of DEAD_{hex}).
5. Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1811 The confidence word at **RAM/ROM** <32> address 20000_{hex} (U410/U420) was not correct.

Error Index E1812 The confidence word at **RAM/ROM** <32> address 20002_{hex} (U410/U420) was not correct.

Caveats The confidence words, which are used for the NVRAM in both bank 0 and bank 2 of the memory map, are initialized by the normal operating firmware in the system. So, if the instrument has never entered normal operating mode, this test will fail. Once initialized, the confidence words should remain complementary as long as the **Battery Backup** batteries <31> are good.

Executive

Routine Name Data Lines 0

Overview This test verifies the data lines to the battery backed-up RAM/ROM <32> in bank 0 (of the Executive memory map) by performing a "walking one's" test on RAM location 20000hex (U410/U420).

CAUTION

Turning the instrument power off during the execution of this test may result in losing some of the battery backed-up data used in normal operation and cause an NVRAM Banks battery test failure to occur. Operation should not be seriously affected, however.

Description

1. Store the current bank selection by reading Bank Decode/Select U530A <27>.
2. Select RAM/ROM <32> bank 0 by setting Bank Decode/Select U520-2 & 5 <27> low.
3. Perform a "walking one's" test on RAM/ROM address 20000hex.
 - Write the pattern 0001hex to RAM/ROM address 20000hex. Read the same address and verify that it was 0001hex. Continue this write/read/verify sequence with patterns 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, and 8000hex.
4. Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.

Error Index E1821 The pattern read did not match the pattern written. The correspondence of the data bits to RAM devices is shown in the following table:

Data Bits	Device
0 – 7	U410
8 – 15	U420

Caveats Turning the instrument power off during the execution of this test may result in losing the RAM/ROM contents of the location being tested. This will cause the normal operating firmware to reset all of the RAM/ROM contents (thus losing Stored Settings and Save Trace Descriptions) and cause the previous diagnostic battery test to fail.

Routine Name

Addrs/Data 0

Overview

This test verifies the address lines and data integrity of the battery backed-up RAM/ROM <32> in bank 0 (of the Executive memory map) by performing a RAM test on all locations.

CAUTION

Turning the instrument power off during the execution of this test may result in losing some of the battery backed-up data used in normal operation and cause an NVRAM Banks battery test failure to occur. Operation should not be seriously affected, however.

Description

1. Save the current bank selection by reading Bank Decode/Select U530A <27>.
2. Select RAM/ROM <32> bank 0 by setting Bank Decode/Select U520-2 & 5 <27> low.
3. Verify Odd DRAM <34> and Even DRAM <34> address range 40000-4FFFF_{hex} (64k bytes). Terminate test if any verify operation fails.
 - Fill address range 40000-4FFFF_{hex} with the pattern AAAA_{hex}.
 - Read and verify address 40000_{hex} for AAAA_{hex}. If so, write CCCC_{hex} to address 40000_{hex}. Increment the address and continue this write/read/verify sequence until address 4FFFF_{hex} is reached.
 - Repeat the read/verify/write sequence, starting again at address 40000_{hex}, for CCCC_{hex}, 5555_{hex}, and 0000_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern).
4. If Odd DRAM <34> and Even DRAM <34> address range 40000-4FFFF_{hex} is free of faults, then verify RAM/ROM <32> address range 20000-2FFFF_{hex}.
 - Save the two RAM/ROM battery confidence words starting at address 20000_{hex} in bank 0 and then set them to BAD1_{hex} and BAD2_{hex}, respectively (in case power gets turned off).
 - Save data in RAM/ROM address range 20000-2FFFF_{hex} (64k bytes) by copying it to Odd DRAM and Even DRAM address range 40000-4FFFF_{hex}.
 - Perform the same procedure as in step 3 for RAM/ROM address range 20000-2FFFF_{hex}, using patterns AAAA_{hex}, CCCC_{hex}, F0F0_{hex}, 5555_{hex}, and AAAA_{hex}.
 - Make one last read/verify pass for the pattern AAAA_{hex} starting at address 20000_{hex}.

Executive

- Copy RAM/ROM data from Odd DRAM and Even DRAM address range 40000-4FFFF_{hex} to RAM/ROM address range 20000-2FFFF_{hex}. In order to restore the RAM/ROM confidence words last, the copying is done from the last address to the first (i.e., from 2FFFE_{hex} to 20000_{hex}).
 - Restore the saved RAM/ROM confidence words to address 20000_{hex} in bank 0.
5. If step 4 passed, then verify RAM/ROM <32> address range 30000-3BFFF_{hex} (48k bytes) in a similar manner as in step 4.
 6. Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.

Error Index E1831

The pattern read from Odd DRAM <34> and Even DRAM <34>, or the pattern read from bank 0 battery backed-up RAM/ROM <32>, was not the pattern written.

If the displayed memory location is between 20000-3BFFF_{hex}, then the correspondence of the data bits to RAM/ROM <32> devices is shown in the following table:

Address(hex)	D0—D7	D8—D15
20000—2FFFF	U410	U420
30000—3BFFF	U440	U430

See Also

Executive Subsys Comm Waveform RAM Address/Data (E523X) if the faulty address was between 40000-4FFFF_{hex}.

Caveats

Turning the instrument power off during the execution of this test may result in losing the RAM/ROM contents of some or all of the locations being tested. This will cause the normal operating firmware to reset all of the RAM/ROM contents (thus losing Stored Settings and Save Trace Descriptions) and cause the previous diagnostic battery test to fail.

Memory

NVRAM Banks

Data Lines 2 (E184X)

Routine Name

Data Lines 2

Overview

This test verifies the data lines to the battery backed-up RAM/ROM <32> in bank 2 (of the Executive memory map) by performing a "walking one's" test on RAM location 20000hex (U510/U520).

CAUTION

Turning the instrument power off during the execution of this test may result in losing some of the battery backed-up data used in normal operation and cause an NVRAM Banks battery test failure to occur. Operation should not be seriously affected, however.

Description

1. Store the current bank selection by reading **Bank Decode/Select** U530A <27>.
2. Select **RAM/ROM** <32> bank 2 by setting **Bank Decode/Select** U520-2 low and U520-5 high <27>.
3. Perform a "walking one's" test on **RAM/ROM** address 20000hex.
 - Write the pattern 0001hex to **RAM/ROM** address 20000hex. Read the same address and verify that it was 0001hex. Continue this write/read/verify sequence with patterns 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, and 8000hex.
4. Restore the bank selection by writing the stored bank selection to **Bank Decode/Select** U520 <27>.

Error Index E1841

The pattern read did not match the pattern written. The correspondence of the data bits to RAM devices is shown in the following table:

Data Bits	Device
0 – 7	U510
8 – 15	U520

Caveats

Turning the instrument power off during the execution of this test may result in losing the **RAM/ROM** contents of the location being tested. This will cause the normal operating firmware to reset all of the **RAM/ROM** contents (thus losing Stored Settings and Save Trace Descriptions) or cause the previous diagnostic battery test to fail.

Executive

Routine Name	Addr/Data 2
--------------	-------------

Overview	This test verifies the address lines and data integrity of the battery backed-up RAM/ROM <32> in bank 2 (of the Executive memory map) by performing a RAM test on all locations.
----------	--

CAUTION

Turning the instrument power off during the execution of this test may result in losing some of the battery backed-up data used in normal operation and cause an NVRAM Banks battery test failure to occur. Operation should not be seriously affected, however.

Description	<ol style="list-style-type: none"> 1. Save the current bank selection by reading Bank Decode/Select U530A <27>. 2. Select RAM/ROM <32> bank 2 by setting Bank Decode/Select U520-2 low and U520-5 high <27>. 3. Verify Odd DRAM <34> and Even DRAM <34> address range 40000-4FFFF_{hex} (64k bytes). Terminate test if any verify operation fails. <ul style="list-style-type: none"> • Fill address range 40000-4FFFF_{hex} with the pattern AAAA_{hex}. • Read and verify address 40000_{hex} for AAAA_{hex}. If so, write CCCCH_{hex} to address 40000_{hex}. Increment the address and continue this write/read/verify sequence until address 4FFFF_{hex} is reached. • Repeat the read/verify/write sequence, starting again at address 40000_{hex}, for CCCCH_{hex}, 5555_{hex}, and 0000_{hex} (i.e., reading and verifying the previous pattern written and then writing the next pattern). 4. If Odd DRAM <34> and Even DRAM <34> address range 40000-4FFFF_{hex} is free of faults, then verify RAM/ROM <32> address range 20000-2FFFF_{hex}. <ul style="list-style-type: none"> • Save the two RAM/ROM battery confidence words starting at address 20000_{hex} in bank 0 and then set them to BAD1_{hex} and BAD2_{hex}, respectively (in case power gets turned off). • Save data in RAM/ROM address range 20000-2FFFF_{hex} (64k bytes) by copying it to Odd DRAM and Even DRAM address range 40000-4FFFF_{hex}. • Perform the same procedure as in step 3 for RAM/ROM address range 20000-2FFFF_{hex}, using patterns AAAA_{hex}, CCCCH_{hex}, F0F0_{hex}, 5555_{hex}, and AAAA_{hex}. • Make one last read/verify pass for the pattern AAAA_{hex} starting at address 20000_{hex}.
-------------	---

Executive

- Copy RAM/ROM data from Odd DRAM and Even DRAM address range 40000-4FFFF_{hex} to RAM/ROM address range 20000-2FFFF_{hex}. In order to restore the RAM/ROM confidence words last, the copying is done from the last address to the first (i.e., from 2FFFE_{hex} to 20000_{hex}).
 - Restore the saved RAM/ROM confidence words to address 20000_{hex} in bank 0.
5. If step 4 passed, then verify RAM/ROM <32> address range 30000-3BFFF_{hex} (48k bytes) in a similar manner as in step 4.
6. Restore the bank selection by writing the stored bank selection to Bank Decode/Select U520 <27>.

Error Index E1851

The pattern read from Odd DRAM <34> and Even DRAM <34>, or the pattern read from bank 2 battery backed-up RAM/ROM <32>, was not the pattern written.

If the displayed memory location is between 20000-3BFFF_{hex}, then the correspondence of the data bits to RAM/ROM <32> devices is shown in the following table:

Address(hex)	D0—D7	D8—D15
20000—2FFFF	U510	U520
30000—3BFFF	U540	U530

See Also Executive Subsys Comm Waveform RAM Address/Data (E523X) if the faulty address was between 40000-4FFFF_{hex}.

Caveats Turning the instrument power off during the execution of this test may result in losing the RAM/ROM contents of some or all of the locations being tested. This will cause the normal operating firmware to reset all of the RAM/ROM contents (thus losing Stored Settings and Save Trace Descriptions) and cause the previous diagnostic battery test to fail.

Routine Name Master

Overview This test verifies the **Interrupt Controllers MASTER U350 <28> Mask Register**. Each bit in the register is set high and low to verify the interrupt masking capability of this device.

Description

1. Disable external interrupts at the processor by setting processor interrupt enable flag low.
2. Perform a "walking one's" test on the MASTER Mask Register.
 - Write, read, and verify patterns *01hex*, *02hex*, *04hex*, *08hex*, *10hex*, *20hex*, *40hex*, and *80hex*.
3. Mask all MASTER interrupts by writing *FFhex* to its Mask Register.
4. Enable external interrupts at the processor by setting processor interrupt enable flag high.

Error Index E2111 The value read did not match the value written. The Mask Register's bits correspond to the following interrupts:

Bit 0 : Not used
 Bit 1 : **Interrupt Controllers SLAVE 1 U360 <28>**
 Bit 2 : **Timer U822 <25> COUNTER 0**
 Bit 3 : **Interrupt Controllers SLAVE 3 U370 <28>**
 Bit 4 : **Front Panel Controller U103 <3>**
 Bit 5 : Not used
 Bit 6 : **Printer Controller U430 <6>**
 Bit 7 : Not used

If the actual value reads zero, then the interrupt represented by the set bit in the expected value is unmaskable.

If the actual value has more than one bit set, then the interrupt represented by the extra set bit(s) in the actual value is masked all the time.

Routine Name Slave 1

Overview This test verifies the **Interrupt Controllers SLAVE 1 U360 <28> Mask Register**. Each bit in the register is set high and low to verify the interrupt masking capability of this device.

Description

1. Disable external interrupts at the processor by setting the processor interrupt enable flag low.
2. Perform a "walking one's" test on the SLAVE 1 Mask Register.
 - Write, read, and verify patterns *01hex*, *02hex*, *04hex*, *08hex*, *10hex*, *20hex*, *40hex*, and *80hex*.
3. Mask all SLAVE 1 interrupts by writing *FFhex* to its Mask Register.
4. Enable external interrupts at the processor by setting processor interrupt enable flag high.

Error Index E2121 The value read did not match the value written. The Mask Register's bits correspond to the following interrupts:

Bit 0 : **MMU Gate Array U210 <33>** sequential address generator
 Bit 1 : **MMU Gate Array U210 <33>** digitizer end message
 Bit 2 : **MMU Gate Array U210 <33>** display talk request
 Bit 3 : Not used
 Bit 4 : **Serial Data Interface U330 <25>**
 Bit 5 : Not used
 Bit 6 : **DMA Controller U800 <28>**
 Bit 7 : Not used

If the actual value reads zero, then the interrupt represented by the set bit in the expected value is unmaskable.

If the actual value has more than one bit set, then the interrupt represented by the extra set bit(s) in the actual value is masked all the time.

Routine Name Slave 3

Overview This test verifies the **Interrupt Controllers SLAVE 3 U370 <28> Mask Register**. Each bit in the register is set high and low to verify the interrupt masking capability of this device.

Description

1. Disable external interrupts at the processor by setting processor interrupt enable flag low.
2. Perform a "walking one's" test on the SLAVE 3 Mask Register.
 - Write, read, and verify patterns *01hex*, *02hex*, *04hex*, *08hex*, *10hex*, *20hex*, *40hex*, and *80hex*.
3. Mask all the SLAVE 3 interrupts by writing *FFhex* to its Mask Register.
4. Enable external interrupts at the processor by setting processor interrupt enable flag high.

Error Index E2131 The value read did not match the value written. The Mask Register's bits correspond to the following interrupts:

Bit 0 : **Timer U822 <25> COUNTER 1**
Bit 1 : **Timer U822 <25> COUNTER 2**
Bit 2 : **GPIB Controller U410 <5>**
Bit 3 : Not used
Bit 4 : **Std RS232 Controller U311 <5>**
Bit 5 : **Optional RS232 Controller U331 <5>**
Bit 6 : **Real Time Clock U614 <25>**
Bit 7 : Not used

If the actual value reads zero, then the interrupt represented by the set bit in the expected value is unmaskable.

If the actual value has more than one bit set, then the interrupt represented by the extra set bit(s) in the actual value is masked all the time.

Control

Timers

Timer 0 (E221X)

Routine Name Timer 0

Overview This test verifies the **Timer U822 COUNTER 0** <25> and the **Clock Generator** <25> by counting 2MHz clock for a short duration.

Description

1. Program the **Timer COUNTER 0** to count down with *6464hex* as the starting count.
 - Write *34hex*, *64hex*, and *64hex* to **Timer Control Word Register** and **Count Register low and high bytes**, respectively.
2. Perform a software delay.
3. Stop the **Timer COUNTER 0** by programming its **Control Word Register** with *34hex*.
4. Read and verify the **Count Register high and low bytes**.
5. Restore the **Timer COUNTER 0** for the normal operating firmware.

Error Index E2211 The **Timer COUNTER 0** counted faster than expected.

Error Index E2212 The **Timer COUNTER 0** counted slower than expected.

Caveats This test may fail if the **Main Processor Board Clock Generator** <27> reference Y950 is out of tolerance since it is being used as the timing reference for the test via the software delay mechanism.

Executive

Routine Name Timer 1

Overview This test verifies the **Timer** U822 COUNTER 1 <25> by counting 6 MHz clock for a short duration.

Description

1. Configure the **Timer** COUNTER 1 to count 6 MHz clock.
 - Write *00hex* to **Timer Configuration Logic** U720 <25> so that G1 U822-14 is high and PCLK(H) connected to CLK1 U822-15.
2. Program the counter to count down with *6464hex* as the starting count.
 - Write *70hex*, *64hex*, and *64hex* to **Timer** Control Word Register and Count Register low and high bytes, respectively.
3. Perform a software delay.
4. Stop the **Timer** COUNTER 1 by programming its Control Word Register with *70hex*.
5. Read and verify the Count Register low and high bytes.

Error Index E2221 The **Timer** COUNTER 1 counted faster than expected.

Error Index E2222 The **Timer** COUNTER 1 counted slower than expected.

Control

Timers

Timer 2 (E223X)

Routine Name Timer 2

Overview This test verifies the **Timer U822 COUNTER 2** <25> counting capability by using the gate control to stop and resume counting.

Description

1. Configure the **Timer COUNTER 2** to count 6 MHz clock.
 - Write *00hex* to **Timer Configuration Logic U720** <25> so that G2 U822-16 is high and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with *6464hex* as the starting count.
 - Write *B0hex*, *64hex*, and *64hex* to **Timer Control Word Register** and **Count Register** low and high bytes, respectively.
3. Perform a software delay.
4. Disable the counter.
 - Write *40hex* to **Timer Configuration Logic U720** <25> so that G2 U822-16 is low.
5. Perform a software delay.
6. Enable the counter.
 - Write *00hex* to **Timer Configuration Logic U720** <25> so that G2 U822-16 is high.
7. Perform a software delay.
8. Stop the **Timer COUNTER 2** by programming its **Control Word Register** with *B0hex*.
9. Read and verify the **Count Register** low and high bytes.

Error Index E2231 The **Timer COUNTER 2** either does not count correctly or its gate has no effect on its counting.

Executive

Routine Name Diagn Signal

Overview This test verifies the **Timer U822 COUNTER 2** <25> external event counting capability through **DIAGNSIG(L)** by using the **Wait State Generator** <28> and the **Wait State Diagnostic Enable** <28> to generate a known number of one wait states.

Description

1. Configure the **Timer COUNTER 2** to count **DIAGNSIG(L)**.
 - Write *80hex* to **Timer Configuration Logic U720** <25> so that **G2 U822-16** is high and **DIAGNSIG(L)** is connected to **CLK2 U822-18**.
2. Program the counter to count down with *6464hex* as the starting count.
 - Write *B0hex*, *64hex*, and *64hex* to **Timer Control Word Register** and **Count Register low** and **high bytes**, respectively.
3. Enable the **Wait State Diagnostic Enable's** **DIAGNSIG(L)** <28> signal to the Executive Bus by writing *01hex* to **U110B** <28>.
4. Generate *100hex* one wait states by repeatedly writing to the **Timer** <25> **COUNTER 2 Control Word Register**. Each write to the **Timer** generates one wait state through **Wait State Generator** <28>. Each wait state generated causes **DIAGNSIG(L)** to toggle.
5. Disable the **Wait State Diagnostic Enable's** **DIAGNSIG(L)** <28> to the Executive Bus by writing *00hex* to **U110B** <28>.
6. Stop the **Timer COUNTER 2** by programming its **Control Word Register** with *B0hex*.
7. Read and verify the **Count Register low** and **high bytes**.

Error Index E2241 The counter counted more external events than expected.

Error Index E2242 The counter counted less external events than expected.

Control

TimerIntrpts

Timer 0 (E231X)

Routine Name Timer 0

Overview This test verifies the **Timer U822 <25> COUNTER 0** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Clear the **Timer COUNTER 0** interrupt.
 - Write *00hex* to **Timer Configuration Logic U720 <25>**, *34hex* to **Timer U822 <25> Control Word Register**, and then toggle **Timer 0 Interrupt Reset U812B-13 <25>**.
2. Verify that there is no **Timer COUNTER 0** interrupt pending at the **Interrupt Controllers MASTER U350 <28>**.
 - Read **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that **IR2 U350-20 <28>** is low.
3. If there is no **Timer COUNTER 0** interrupt pending, then enable **Timer COUNTER 0** interrupt by setting bit 2 of the **Interrupt Controllers MASTER U350 <28> Mask Register** low.
4. Complete the programming for **Timer COUNTER 0** with *80hex* as the starting count.
 - Write *80hex* and *00hex* to **Count Register low and high bytes**. This starts the counter.
5. Perform a software delay.
6. Verify that the **Timer COUNTER 0** interrupt did occur.
7. Disable the **Timer COUNTER 0** interrupt by setting bit 2 of the **Interrupt Controllers MASTER U350 <28> Mask Register** high.
8. Restore the **Timer COUNTER 0** for the normal operating firmware.

Error Index E2311 The **Timer COUNTER 0** interrupt did not occur.

Error Index E2312 The **Timer COUNTER 0** interrupt could not be cleared at the **Interrupt Controllers MASTER U350**.

Executive

Control	TimerIntrpts	Timer 1 (E232X)
Routine Name	Timer 1	
Overview	This test verifies the Timer U822 <25> COUNTER 1 interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.	
Description	<ol style="list-style-type: none"> 1. Configure the Timer COUNTER 1 to count 6 MHz clock. <ul style="list-style-type: none"> • Write <i>00hex</i> to Timer Configuration Logic U720 <25> so that G1 U822-14 is high and PCLK(H) is connected to CLK1 U822-15. 2. Clear Timer COUNTER 1 interrupt. <ul style="list-style-type: none"> • Write <i>70hex</i> to the Control Word Register. 3. Verify that there is no Timer COUNTER 1 interrupt pending at the Interrupt Controllers SLAVE 3 U370 <28>. <ul style="list-style-type: none"> • Read Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER and verify that IR0 U370-18 is low. 4. If there is no Timer COUNTER 1 interrupt pending, then enable the Timer COUNTER 1 interrupt by setting bit 0 of Interrupt Controllers SLAVE 3 U370 <28> Mask Register and bit 3 of Interrupt Controllers MASTER U350 <28> Mask Register low. 5. Complete programming the Timer COUNTER 1 with <i>80hex</i> as the starting count. <ul style="list-style-type: none"> • Write <i>80hex</i> and <i>00hex</i> to the Count Register low and high bytes. This starts the counter. 6. Perform a software delay. 7. Verify that Timer COUNTER 1 interrupt did occur. 8. Disable the Timer COUNTER 1 interrupt by setting bit 0 of Interrupt Controllers SLAVE 3 U370 <28> Mask Register and bit 3 of Interrupt Controllers MASTER U350 <28> Mask Register high. 	
Error Index E2321	The Timer COUNTER 1 interrupt did not occur.	
Error Index E2322	The Timer COUNTER 1 interrupt could not be cleared at the Interrupt Controllers SLAVE 3 U370 .	

Executive

Control

TimerIntrpts

Timer 2 (E233X)

Routine Name Timer 2

Overview This test verifies the **Timer U822 <25> COUNTER 2** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Configure the **Timer COUNTER 2** to count 6 MHz clock.
 - Write *00hex* to **Timer Configuration Logic U720 <25>** so that G2 U822-16 is high and PCLK(H) is connected to CLK2 U822-18.
2. Clear **Timer COUNTER 2** interrupt.
 - Write *B0hex* to the **Control Word Register**.
3. Verify that there is no **Timer COUNTER 2** interrupt pending at the **Interrupt Controllers SLAVE 3 U370 <28>**.
 - Read **Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER** and verify that IR0 U370-19 is low.
4. If there is no **Timer COUNTER 2** interrupt pending, then enable the **Timer COUNTER 2** interrupt by setting bit 1 of **Interrupt Controllers SLAVE 3 U370 <28> Mask Register** and bit 3 of **Interrupt Controllers MASTER U350 <28> Mask Register** low.
5. Complete programming the **Timer COUNTER 2** with *80hex* as the starting count.
 - Write *80hex* and *00hex* to the **Count Register** low and high bytes. This starts the counter.
6. Perform a software delay.
7. Verify that **Timer COUNTER 2** interrupt did occur.
8. Disable the **Timer COUNTER 2** interrupt by setting bit 1 of **Interrupt Controllers SLAVE 3 U370 <28> Mask Register** and bit 3 of **Interrupt Controllers MASTER U350 <28> Mask Register** high.

Error Index E2331 The **Timer COUNTER 2** interrupt did not occur.

Error Index E2332 The **Timer COUNTER 2** interrupt could not be cleared at the **Interrupt Controllers SLAVE 3 U370**.

Executive

Routine Name Zero Wait

Overview This test verifies the **Wait State Generator's** <28> ability to generate zero wait states. The **Timer U822 COUNTER 2** <25> is used to count the number of wait states.

Description

1. Configure the **Timer COUNTER 2** <25> to count 6 MHz clock.
 - Write *40hex* to **Timer Configuration Logic U720** <25> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with *6464hex* as the starting count.
 - Write *B0hex*, *64hex*, and *64hex* to **Timer Control Word Register** and **Count Register** low and high bytes, respectively.
3. Enable the **Wait State Diagnostic Enable** circuit by writing *01hex* to U110B <28> so that U411B-9 <28> is gated onto DIAGNSIG(L) <25>.
4. Generate *100hex* zero wait requests by repeatedly writing *00hex* to **Memory Diagnostic Signal Select U352B** <29>. The **Wait State Generator** sets DIAGNSIG(L) low when it is generating wait states. The **Timer COUNTER 2** is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflects the number of wait states generated.
5. Disable the **Wait State Diagnostic Enable** circuit by writing *00hex* to U110B <28> so that U411B-9 <28> is no longer gated onto DIAGNSIG(L) <25>.
6. Stop the **Timer COUNTER 2** by programming its **Control Word Register** with *B0hex*.
7. Read and verify the **Timer COUNTER 2 Count Register** low and high bytes.

Error Index E2411 The number of wait states was more than zero wait states.

Routine Name One Wait

Overview This test verifies the **Wait State Generator's** <28> ability to generate one wait states. The **Timer COUNTER 2** U822 <25> is used to count the number of wait states generated.

Description

1. Configure the **Timer COUNTER 2** <25> to count 6 MHz clock.
 - Write 40hex to **Timer Configuration Logic** U720 <25> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464hex as the starting count.
 - Write B0hex, 64hex, and 64hex to **Timer Control Word Register** and **Count Register** low and high bytes, respectively.
3. Enable the **Wait State Diagnostic Enable** circuit by writing 01hex to U110B <28> so that U411B-9 <28> is gated onto DIAGNSIG(L) <25>.
4. Generate 100hex one wait requests by repeatedly writing 70hex to **Timer COUNTER 1 Control Word Register**. Since the **Timer** device is a one wait state device, each access to it will generate a one wait state. The **Wait State Generator** sets DIAGNSIG(L) low when it is generating wait states. The **Timer COUNTER 2** is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflects the number of wait states generated.
5. Disable the **Wait State Diagnostic Enable** circuit by writing 00hex to U110B <28> so that U411B-9 <28> is no longer gated onto DIAGNSIG(L) <25>.
6. Stop the **Timer COUNTER 2** by programming its **Control Word Register** with B0hex.
7. Read and verify the **Timer COUNTER 2 Count Register** low and high bytes.

Error Index E2421 The **Wait State Generator** generated more than one wait states for the **Timer** <25> device.

Routine Name Two Wait

Overview This test verifies the Wait State Generator's <28> ability to generate two wait states. The Timer COUNTER 2 U822 <25> is used to count the number of wait states.

Description

1. Configure the Timer COUNTER 2 <25> to count 6 MHz clock.
 - Write 40hex to Timer Configuration Logic U720 <25> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with 6464hex as the starting count.
 - Write B0hex, 64hex, and 64hex to Timer Control Word Register and Count Register low and high bytes, respectively.
3. Enable the Wait State Diagnostic Enable circuit by writing 01hex to U110B <28> so that U411B-9 <28> is gated onto DIAGNSIG(L) <25>.
4. Generate 100hex two wait requests by repeatedly writing 60hex to the Front Panel Controller U103 <3>. Since the Front Panel Controller device is a two wait state device each access to it will generate two wait states. The Wait State Generator sets DIAGNSIG(L) low when it is generating wait states. The Timer COUNTER 2 is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflects the number of wait states generated.
5. Disable the Wait State Diagnostic Enable circuit by writing 00hex to U110B <28> so that U411B-9 <28> is no longer gated onto DIAGNSIG(L) <25>.
6. Stop the Timer COUNTER 2 by writing B0hex to its Control Word Register.
7. Read and verify the Timer COUNTER 2 Count Register low and high bytes.

Error Index E2431 The Wait State Generator generated more than two wait states for the Front Panel Controller <3> device.

Control	MPU Waits	Four Wait (E244X)
---------	-----------	-------------------

Routine Name	Four Wait
--------------	-----------

Overview	This test verifies the Wait State Generator's <28> ability to generate four wait states. The Timer COUNTER 2 U822 <25> is used to count the number of wait states.
----------	--

Description	<ol style="list-style-type: none"> 1. Configure the Timer COUNTER 2 <25> to count 6 MHz clock. <ul style="list-style-type: none"> • Write 40hex to Timer Configuration Logic U720 <25> so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18. 2. Program the counter to count down with 6464hex as the starting count. <ul style="list-style-type: none"> • Write B0hex, 64hex, and 64hex to Timer Control Word Register and Count Register low and high bytes, respectively. 3. Enable the Wait State Diagnostic Enable circuit by writing 01hex to U110B <28> so that U411B-9 <28> is gated onto DIAGNSIG(L) <25>. 4. Generate 100hex four wait requests by repeatedly writing A6hex to the Printer Controller U430 <6> Control Word Register. Since the Printer Controller is a four wait state device, each access to it will generate four wait states. The Wait State Generator sets DIAGNSIG(L) low when it is generating wait states. The Timer COUNTER 2 is setup to count 6 MHz clock when DIAGNSIG(L) is low and hence the count reflect the number of wait states generated. 5. Disable the Wait State Diagnostic Enable circuit by writing 00hex to U110B <28> so that U411B-9 <28> is no longer gated onto DIAGNSIG(L) <25>. 6. Stop the Timer COUNTER 2 by programming its Control Word Register with B0hex. 7. Read and verify the Timer COUNTER 2 Count Register low and high bytes.
-------------	---

Error Index E2441	The Wait State Generator generated more than four wait states for the Printer Controller <6> device.
-------------------	--

Caveats	There is no signal to represent a four wait state request coming into the Wait State Generator <28>. This is the default maximum number of wait states generated by this circuit.
---------	---

Control

ROM Waits

Zero Wait (E251X)

Routine Name Zero Wait

Overview This test verifies the **Memory Wait State Gen <29>** circuit on the Memory Board. The **Timer U822 <25> COUNTER 2** is used to count the number of wait states.

Description

1. Configure the **Timer COUNTER 2 <25>** to count 6 MHz clock.
 - Write *40hex* to **Timer Configuration Logic U720 <25>** so that G2 U822-16 is low and PCLK(H) is connected to CLK2 U822-18.
2. Program the counter to count down with *6464hex* as the starting count.
 - Write *B0hex*, *64hex*, and *64hex* to **Timer Control Word Register** and **Count Register** low and high bytes, respectively.
3. Enable the **Memory Diagnostic Signal Select** circuit by writing *01hex* to U352B <29> so that U820B-9 <28> is gated onto DIAGNSIG(L) <25>.
4. Generate *100hex* zero wait requests by repeatedly reading an EPROM <28> location *D0000hex*. Each time the EPROM is selected, **Memory Wait State Gen <29>** generates zero wait states. The **Memory Wait State Gen** sets DIAGNSIG(L) low when it is generating wait states. As the **Timer COUNTER 2** is set up to count 6 MHz clock (wait states are specified in terms of the PCLK(H) cycles which is at 6 MHz) during the time DIAGNSIG(L) is low, the number of wait states generated by the **Memory Wait State Gen** is reflected in the counter.
5. Disable the **Memory Diagnostic Signal Select** circuit by writing *00hex* to U352B <28> so that U820B-9 <28> is no longer gated to DIAGNSIG(L) <25>.
6. Stop the **Timer COUNTER 2** by programming its **Control Word Register** with *B0hex*.
7. Read and verify the **Timer COUNTER 2 Count Register** low and high bytes.

Error Index E2511 The **Memory Wait State Gen** generated more than zero wait states.

Executive

Routine Name Floating Pt (Floating Point)

Overview This test verifies the floating point and transcendental function capabilities of the **Numeric Processor Extension U500 <27>**.

Description 1. Compute the following equation using the **Numeric Processor Extension <27>** and verify that the result is within a tolerance of ± 0.005 of the expected value.

$$\frac{\log_e(15.3) * \log_{10}(23.5) * \exp(0.12)}{\text{atan}[\sin(0.7071) * \cos(0.5) * \tan(0.5774)]}$$

Error Index E2611 The result from the floating point calculation was not 11.8582 ± 0.005 .

Routine Name DMA 0

Overview This test verifies the **DMA Controller** <28> channel 0 by performing a memory-to-memory DMA transfer in **DRAM** <30>.

Description

1. Check to see if the **DMA Controller** is accessible.
 - Write, read, and verify `7F01hex` to the General Mode Register of the **DMA Controller**.
2. If the **DMA Controller** is accessible (i.e., `7F01hex` was verified in step 1), construct the command block for the transfer at **DRAM** <30> address `30000hex`. Initialize the source and destination memory locations.
 - Write `C8DDhex`, `0013hex`, `0020hex`, `0013hex`, `0040hex`, `0000hex`, `0010hex`, `0000hex`, and `0401hex` to locations starting from `30000hex`.
 - Initialize the source **DRAM** address range `30020hex`-`3002Ahex` with patterns `AAAAhex`, `CCCChex`, `F0F0hex`, `FF00hex`, and `5555hex`.
 - Initialize the destination **DRAM** address range `30040hex`-`30049hex` with pattern `8000hex`.
3. Program the **DMA Controller** channel 0 to read the command block from memory and start the transfer of five words from `30020hex` to `30040hex`.
 - Write `7F01hex`, `00hex`, `00hex`, `0013hex`, `0000hex`, and `001Ahex` to General Mode Register, General Burst Register, General Delay Register, Command Pointer Register High, Command Pointer Register Low and General Command Register, respectively.
4. Perform a software delay.
5. Clear the DMA interrupt at the **DMA Controller** and verify that the transfer was successful.
 - Write `0018hex` General Command Register.
 - Compare the contents of the destination and source memory locations and verify that they are the same.

Error Index E2711 The value read from the DMA Controller's General Mode Register did not match what was written.

Error Index E2712 The DMA transfer was not successful. The display shows the first pattern that did not verify.

Control

DMA

Interrupt (E272X)

Routine Name Interrupt

Overview This test verifies the **DMA Controller <28>** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated by performing a memory-to-memory DMA transfer.

Description

1. Check to see if the **DMA Controller** is accessible.
 - Write, read, and verify *7F01hex* to the **General Mode Register** of the **DMA Controller**.
2. If the **DMA Controller** is accessible (i.e., *7F01hex* is verified in the step 1), program the **DMA Controller** channel 0 and clear its interrupt, if any.
 - Write *7B01hex*, *00hex*, *00hex*, *0013hex*, *0000hex*, and *0018hex* to **General Mode Register**, **General Burst Register**, **General Delay Register**, **Command Pointer Register High**, **Command Pointer Register Low** and **General Command Register**, respectively.
3. Verify that there is no DMA interrupt pending at the **Interrupt Controllers SLAVE 1 U360 <28>**.
 - Read **Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER** and verify that **IR6 U360-24** is low.
4. If there is no DMA interrupt pending, construct the command block for the transfer at **DRAM** address *30000hex*. Initialize the source and destination memory locations.
 - Write *C8DDhex*, *0013hex*, *0020hex*, *0013hex*, *0040hex*, *0000hex*, *0010hex*, *0000hex*, and *0401hex* to locations starting from *30000hex*.
 - Initialize the source **DRAM** address range *30020hex*-*3002Ahex* with patterns *AAAAhex*, *CCCChex*, *F0F0hex*, *FF00hex*, and *5555hex*.
 - Initialize the destination **DRAM** address range *30040hex*-*30049hex* with pattern *8000hex*.
5. Enable the DMA interrupt by setting bit 6 of **Interrupt Controllers SLAVE 1 U360 <28> Mask Register** low and bit 1 of **Interrupt Controllers MASTER U350 <28> Mask Register** low.
6. Start the DMA transfer by writing *1Ahex* to **General Command Register**.
7. Perform a software delay.
8. Verify that the DMA interrupt occurred.
9. Disable the DMA interrupt by setting bit 6 of **Interrupt Controllers SLAVE 1 U360 <28> Mask Register** high and bit 1 of **Interrupt Controllers MASTER U350 <28> Mask Register** high.

Executive

Control

DMA's

Interrupt (E272X)

Error Index E2721	The value read from the DMA Controller's General Mode Register did not match what was written.
Error Index E2722	The DMA interrupt did not occur.
Error Index E2723	The DMA interrupt could not be cleared at Interrupt Controllers SLAVE 1 U360.

Executive

Front Panel

Control

RAM (E311X)

Routine Name RAM

Overview This test verifies the **Front Panel Controller** U103 display RAM <3> by performing a RAM test on the display RAM locations.

Description

1. Initialize the **Front Panel Controller** <3>.
 - Write *04hex* and *22hex* to the Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
2. Save the contents of the **Front Panel Controller** display RAM <3>.
 - Write *70hex* to the Command Register and read eight bytes from the Data Register.
3. Verify the RAM locations.
 - Fill all eight locations with the pattern *AAhex*.
 - Read and verify location 0 for *AAhex*. If so, write *CChex* to location 0. Increment the address and continue the read/verify/write sequence until location 7 is reached.
 - Read and verify location 7 for *CChex*. If so, write *F0hex* to location 7. Decrement the address and continue the read/verify/write sequence until location 0 is reached.
 - Repeat the read/verify/write sequence starting at location 0 and 7 for alternative passes, for patterns *F0hex* and *55hex*.
 - Make one last read/verify pass for the pattern *55hex* starting at location 7 and also restore the saved contents of the display RAM.
4. Initialize the **Front Panel Controller** <3>.
 - Write *04hex* and *22hex* to the Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.

Error Index E3111 The pattern read did not match the pattern written from the **Front Panel Controller** display RAM <3>. The display shows the first pattern that did not verify.

Executive

Routine Name	RAM Control
Overview	This test verifies various display RAM controls of the Front Panel Controller U103 <3> by exercising these controls.
Description	<ol style="list-style-type: none">1. Save the contents of the Front Panel Controller display RAM <3>.<ul style="list-style-type: none">• Write <i>70hex</i> to the Command Register and read eight bytes from the Data Register.2. Initialize the Front Panel Controller <3>.<ul style="list-style-type: none">• Write <i>04hex</i> and <i>22hex</i> to the Command Register and then write <i>E0hex</i> 33 times, again to the Command Register, to assure that IRQ U103-4 is low.3. Verify that the display RAM can be cleared.<ul style="list-style-type: none">• Write <i>DDhex</i> to the Command Register and read the Front Panel Controller Status Word Register and verify that the display RAM is unavailable (i.e., bit 7 is high).• Perform a software delay (approximately 300μs).• Read the Front Panel Controller Status Word Register and verify that the display RAM is available (i.e., bit 7 is low).4. Verify that all the RAM locations may be set high.<ul style="list-style-type: none">• Write <i>60hex</i> to the Command Register. Read eight bytes from the Data Register and verify that each of them is <i>FFhex</i>.5. Verify that all the RAM locations may be set low.<ul style="list-style-type: none">• Write <i>D3hex</i> to the Command Register.• Write <i>60hex</i> to the Command Register. Read eight bytes from the Data Register and verify that each of them is <i>00hex</i>.6. Initialize the Front Panel Controller <3>.<ul style="list-style-type: none">• Write <i>04hex</i> and <i>22hex</i> to the Command Register and then write <i>E0hex</i> 33 times, again to the Command Register, to assure that IRQ U103-4 is low.7. Restore the display RAM locations by writing back the stored contents of the display RAM.<ul style="list-style-type: none">• Write <i>90hex</i> to the Command Register and write eight bytes to the Data Register.

Front Panel

Control

RAM Control (E312X)

- Error Index E3121** The **Front Panel Controller** display RAM <3> was available when it should have been unavailable (during the time it was being cleared).
- Error Index E3122** The **Front Panel Controller** display RAM <3> was unavailable when it should have been available (150 μ s after it was cleared).
- Error Index E3123** The **Front Panel Controller** display RAM <3> could not be set to ones, i.e., writing *DD_{hex}* to the Command Register did not set the display RAM locations to ones.
- Error Index E3124** The **Front Panel Controller** display RAM <3> could not be set to zeros, i.e., writing *D3_{hex}* to the Command Register did not set the display RAM locations to zeros.

Front Panel

Control

Interrupt (E313X)

Routine Name Interrupt

Overview This test verifies the **Front Panel Controller** <3> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The infrared LEDs are disabled and enabled to generate the interrupt.

Description

1. Initialize the **Front Panel Controller** <3>.
 - Write *04hex* and *22hex* to the Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.
2. Verify that there is no front panel interrupt pending at the **Interrupt Controllers** <28>.
 - Read **Interrupt Controllers** MASTER U350 <28> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <28> is low.
3. If there was no pending front panel interrupt, enable the front panel interrupt by setting the **Interrupt Controllers** MASTER U350 <28> Mask Register bit 4 low.
4. Disable front panel infrared LEDs by writing *00hex* to **Front Panel Infrared Disable Control** U700B <26> so that IRDIS(L) U212C-11(L) <3> is low.
5. Perform a software delay.
6. Enable front panel infrared LEDs by writing *01hex* to **Front Panel Infrared Disable Control** U700B <26> so that IRDIS(L) U212C-11(L) <3> is high.
7. Perform a software delay.
8. Verify that the front panel interrupt did occur.
9. Disable the front panel interrupt setting **Interrupt Controllers** MASTER U350 Mask Register bit 4 high.
10. Initialize the **Front Panel Controller**.
 - Write *04hex* and *22hex* to the Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 is low.

Error Index E3131 The **Front Panel Controller** <3> interrupt did not occur.

Error Index E3132 The **Front Panel Controller** <3> interrupt could not be cleared at the **Interrupt Controllers** MASTER U350 <28>.

Executive

Front Panel

Hard Keys

Open (E321X)

Routine Name Open

Overview This test verifies the front panel hard keys (buttons) on the Touch Panel <2> and Front Panel Button boards <3> by checking that none of the front panel hard keys is stuck in the closed position.

Description

1. Disable front panel infrared LEDs by writing `00hex` to **Front Panel Infrared Disable Control** U700B <26> so that IRDIS(L) U212C-11 is low.
2. Wait for the front panel infrared LEDs to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).
 - Read the front panel sensor RAM in the **Front Panel Controller** U103 <3> by writing `60hex` to the Command Register and reading eight bytes from the Data Register.
 - Clear the front panel interrupt by writing `E0hex` to the Command Register.
 - Perform a software delay.
 - Read the **Interrupt Controllers** MASTER U350 <28> INTERRUPT REQUEST REGISTER and verify that IR4 U350-22 <28> is low.
 - Repeat the previous four steps until IR4 U350-22 <28> is low.
3. Enable front panel infrared LEDs by writing `01hex` to **Front Panel Infrared Disable Control** U700B <26> so that IRDIS(L) U212C-11 is high.
4. Perform a software delay.
5. Read the sensor RAM in the **Front Panel Controller** U103 <3> after it stabilizes.
 - Use the same procedure outlined in step 2.
6. Verify that the hard keys status in the sensor RAM is open (i.e., hard keys sensor RAM bits are all 1).

Error Index E3211 One or more hard keys was stuck in the closed position.

There are 11 hard keys in the front panel. The state of each of them is represented by a bit in the expected and actual values. A low (0) bit in the actual field represents the hard key (button) that was closed. The following table presents the bit encoding scheme for the hard keys.

Executive

Front Panel

Hard Keys

Open (E321X)

Data Bit (Switch)	Hardkey
bit 0 (S100 <3>)	Acquisition Run/Stop
bit 1 (S200 <3>)	Autoset
bit 2 (S300 <3>)	Hardcopy
bit 3 (S400 <3>)	Sequence Setting
bit 4 (S7 <2>)	Waveform
bit 5 (S6 <2>)	Trigger
bit 6 (S5 <2>)	Measure
bit 7 (S4 <2>)	Store/Recall
bit 8 (S3 <2>)	Enhanced Accuracy
bit 9 (S1 <2>)	Touch Panel On/Off
bit 10 (S2 <2>)	Utility

See Also

The Front Panel Verify Hard Keys test.

Routine Name

Row Open

Overview

This test verifies the 22 front panel infrared emitter/receptor row pairs (y-axis input sensors) in **Touch Panel Bezel Sensors <2>** by checking that all the row receptors are sensing their corresponding emitter.

Description

1. Disable front panel infrared LEDs by writing `00hex` to **Front Panel Infrared Disable Control U700B <26>** so that **IRDIS(L) U212C-11** is low.
2. Wait for the sensor RAM in the **Front Panel Controller U103 <3>** to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).
 - Read the sensor RAM by writing `60hex` to the Command Register and reading eight bytes from the Data Register.
 - Clear the front panel interrupt by writing `E0hex` to the Command Register.
 - Perform a software delay.
 - Read the **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that **IR4 U350-22 <28>** is low.
 - Repeat the previous four steps until **IR4 U350-22 <28>** is low.
3. Enable front panel infrared LEDs by writing `01hex` to **Front Panel Infrared Disable Control U700B <26>** so that **IRDIS(L) U212C-11** is high.
4. Perform a software delay.
5. Read the sensor RAM in the **Front Panel Controller U103 <3>** after it stabilizes.
 - Use the same procedures outlined in step 2.
6. Verify that all the row receptors sense an emitter (i.e., soft key rows sensor RAM bits are 0).

Error Index E3311

One or more row receptors did not sense its corresponding emitter.

There are 22 infrared emitter/receptor row pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A high (1) bit in the actual value represents the receptor that did not sense an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <2>.

Front Panel

Soft Keys

Row Open (E331X)

Set Bit	Emitter/Receptor Pair
bit 0	DS770/CR170
bit 1	DS760/CR160
bit 2	DS761/CR161
bit 3	DS762/CR162
bit 4	DS763/CR150
bit 5	DS750/CR151
bit 6	DS751/CR152
bit 7	DS752/CR153
bit 8	DS740/CR140
bit 9	DS741/CR141
bit 10	DS742/CR142
bit 11	DS743/CR130
bit 12	DS730/CR131
bit 13	DS731/CR132
bit 14	DS732/CR133
bit 15	DS720/CR120
bit 16	DS721/CR121
bit 17	DS722/CR122
bit 18	DS723/CR110
bit 19	DS710/CR111
bit 20	DS711/CR112
bit 21	DS712/CR113

See Also

The Front Panel Verify Soft Keys test.

Executive

Routine Name

Column Open

Overview

This test verifies the 11 front panel infrared emitter/receptor column pairs (x-axis input sensors) in **Touch Panel Bezel Sensors <2>** by checking that all of the column receptors are sensing their corresponding emitters.

Description

1. Disable front panel infrared LEDs by writing `00hex` to **Front Panel Infrared Disable Control U700B <26>** so that IRDIS(L) U212C-11 is low.
2. Wait for the sensor RAM in the **Front Panel Controller U103 <3>** to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).
 - Read the sensor RAM by writing `60hex` to the Command Register and reading eight bytes from the Data Register.
 - Clear the front panel interrupt by writing `E0hex` to the Command Register.
 - Perform a software delay.
 - Read the **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that IR4 U350-22 <28> is low.
 - Repeat the previous four steps until IR4 U350-22 <28> is low.
3. Enable front panel infrared LEDs by writing `01hex` to **Front Panel Infrared Disable Control U700B <26>** so that IRDIS(L) U212C-11 is high.
4. Perform a software delay.
5. Read the sensor RAM in the **Front Panel Controller U103 <3>** after it stabilizes.
 - Use the same procedures outlined in step 2.
6. Verify that all column the receptors sense an emitter (i.e., soft key columns sensor RAM bits are 0).

Error Index E3321

One or more column receptors did not sense its corresponding emitter.

There are 11 infrared emitter/receptor column pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A high (1) bit in the actual value represents the receptor that did not sense an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <2>.

Front Panel

Soft Keys

Column Open (E332X)

Set Bit	Emitter/Receptor Pair
bit 0	DS270/CR200
bit 1	DS271/CR201
bit 2	DS272/CR202
bit 3	DS370/CR300
bit 4	DS371/CR301
bit 5	DS470/CR400
bit 6	DS471/CR401
bit 7	DS570/CR500
bit 8	DS571/CR501
bit 9	DS572/CR502
bit 10	DS670/CR600

See Also The Front Panel Verify Soft Keys test.

Front Panel

Soft Keys

Row Close (E333X)

Routine Name

Row Close

Overview

This test verifies the 22 front panel infrared emitter/receptor row pairs (y-axis input sensors) in **Touch Panel Bezel Sensors <2>** by checking that all the receptors are capable of sensing an input (i.e., absence of the emitter beam).

Description

1. Disable front panel infrared LEDs by writing *00hex* to **Front Panel Infrared Disable Control U700B <26>** so that IRDIS(L) U212C-11 is low.
2. Wait for the sensor RAM in the **Front Panel Controller U103 <3>** to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).
 - Read the sensor RAM by writing *60hex* to the Command Register and reading eight bytes from the Data Register.
 - Clear the front panel interrupt by writing *E0hex* to the Command Register.
 - Perform a software delay.
 - Read the **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that IR4 U350-22 <28> is low.
 - Repeat the previous four steps until IR4 U350-22 <28> is low.
3. Verify that none of the row receptors sense an emitter (i.e., the soft key rows sensor RAM bits are 1).
4. Enable the front panel infrared LEDs by writing *01hex* to **Front Panel Infrared Disable Control U700B <26>** so that IRDIS(L) U212C-11 is high.
5. Wait for the the front panel sensor RAM to stabilize.
 - Use the same procedures outlined in step 2.

Error Index E3331

One or more row receptors sensed an emitter.

There are 22 infrared emitter/receptor row pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A low (0) bit in the actual value represents the receptor that sensed an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <2>.

Executive

Reset Bit	Emitter/Receptor Pair
bit 0	DS770/CR170
bit 1	DS760/CR160
bit 2	DS761/CR161
bit 3	DS762/CR162
bit 4	DS763/CR150
bit 5	DS750/CR151
bit 6	DS751/CR152
bit 7	DS752/CR153
bit 8	DS740/CR140
bit 9	DS741/CR141
bit 10	DS742/CR142
bit 11	DS743/CR130
bit 12	DS730/CR131
bit 13	DS731/CR132
bit 14	DS732/CR133
bit 15	DS720/CR120
bit 16	DS721/CR121
bit 17	DS722/CR122
bit 18	DS723/CR110
bit 19	DS710/CR111
bit 20	DS711/CR112
bit 21	DS712/CR113

See Also

The Front Panel Verify Soft Keys test.

Front Panel

Soft Keys

Column Close (E334X)

Routine Name

Column Close

Overview

This test verifies the 11 front panel infrared emitter/receptor pairs (x-axis input sensors) in **Touch Panel Bezel Sensors** <2> by checking that all the receptors are capable of sensing an input (i.e., absence of an emitter beam).

Description

1. Disable front panel infrared LEDs by writing `00hex` to **Front Panel Infrared Disable Control** U700B <26> so that IRDIS(L) U212C-11 is low.
2. Wait for the sensor RAM in the **Front Panel Controller** U103 <3> to stabilize (a stable condition is defined to be the state wherein there are no front panel interrupts).
 - Read the sensor RAM by writing `60hex` to the Command Register and reading eight bytes from the Data Register.
 - Clear the front panel interrupt by writing `E0hex` to the Command Register.
 - Perform a software delay.
 - Read the **Interrupt Controllers MASTER** U350 <28> **INTERRUPT REQUEST REGISTER** and verify that IR4 U350-22 <28> is low.
 - Repeat the previous four steps until IR4 U350-22 <28> is low.
3. Verify that none of the column receptors sense an emitter (i.e., soft key columns sensor RAM bits are 1).
4. Enable front panel infra-red LEDs by writing `01hex` to **Front Panel Infrared Disable Control** U700B <26> so that IRDIS(L) U212C-11 is high.
5. Wait for the front panel sensor RAM to stabilize.
 - Use the same procedures outlined in step 2.

Error Index E3341

One or more column receptors sensed an emitter.

There are 11 infrared emitter/receptor column pairs in the front panel. The state of each of them is represented by a bit in the expected and actual values. A low (0) bit in the actual value represents the receptor that sensed an emitter. The following table helps to decode the actual value to the emitter/receptor pairs <2>.

Executive

Front Panel

Soft Keys

Column Close (E334X)

Reset Bit	Emitter/Receptor Pair
bit 0	DS270/CR200
bit 1	DS271/CR201
bit 2	DS272/CR202
bit 3	DS370/CR300
bit 4	DS371/CR301
bit 5	DS470/CR400
bit 6	DS471/CR401
bit 7	DS570/CR500
bit 8	DS571/CR501
bit 9	DS572/CR502
bit 10	DS670/CR600

See Also The Front Panel Verify Soft Keys test.

Routine Name Upper Knob

Overview This test verifies the upper knob circuit in the **Serial Data Interface U330 <25>** by checking that the upper knob can be initialized and exercised properly. Exercising the knob is accomplished through special test increment/decrement operations in U330.

Description

1. Initialize knob circuits in the **Serial Data Interface U330 <25>** by writing *80hex*, *02hex*, and *00hex* to Misc. Control Select 1 Register, Misc. Control Select Register and Misc. Control Select Register, respectively.
2. Perform a software delay.
3. Read the Knob Read Enable 0 Register and verify that the value is *3Fhex*.
4. Index the upper knob counter to read *00hex*.
 - Read the Knob Read Enable 0 Register and depending on its value, manually index the counter to read *00hex*. To increment the counter, patterns *80hex*, *A0hex*, *E0hex*, *C0hex*, and *80hex* are written to the Misc. Control Select 1 Register. To decrement the counter patterns *80hex*, *C0hex*, *E0hex*, *A0hex*, and *80hex* are written to the Misc. Control Select 1 Register.
5. Exercise the counter through its full range in both directions and verify that the counter still reads *00hex*.
 - Decrement the counter 64 times (see step 5). Adjust the counter to 0, and increment the counter 64 times. Read the Misc. Control Select 1 Register and verify that it is *00hex*.

Error Index E3411 The upper knob counter does not read *3Fhex* after initialization.

Error Index E3412 The upper knob counter increment/decrement test failed.

See Also The Front Panel Verify Knobs test.

Routine Name Lower Knob

Overview This test verifies the lower knob circuit in the **Serial Data Interface U330 <25>** by checking that the lower knob can be initialized and exercised properly. Exercising the knob is accomplished through special test increment/decrement operations available in U330.

Description

1. Initialize knob circuits in the **Serial Data Interface U330 <25>** by writing *80hex*, *02hex*, and *00hex* to Misc. Control Select 1 Register, Misc. Control Select Register and Misc. Control Select Register, respectively.
2. Perform a software delay.
3. Read the Knob Read Enable 1 Register and verify that the value is *3Fhex*.
4. Index the lower knob counter to read *00hex*.
 - Read the Knob Read Enable 1 Register and depending on its value, manually index the counter to read *00hex*. To increment the counter, patterns *80hex*, *A0hex*, *E0hex*, *C0hex*, and *80hex* are written to the Misc. Control Select 1 Register. To decrement the counter patterns *80hex*, *C0hex*, *E0hex*, *A0hex*, and *80hex* are written to the Misc. Control Select 1 Register.
5. Exercise the counter through its full range in both directions and verify that the counter still reads *00hex*.
 - Decrement the counter 64 times (see step 5). Adjust the counter to 0, and increment the counter 64 times. Read the Knob Read Enable 1 Register and verify that it is *00hex*.

Error Index E3421 The lower knob counter does not read *3Fhex* after initialization.

Error Index E3422 The lower knob counter increment/decrement test failed.

See Also The Front Panel Verify Knobs test.

Routine Name Hard Keys

Overview This test verifies the hard keys by allowing the user to interactively press the hard keys. Both visual and audio feedback is provided.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off." Once this test is invoked, the operator may press any of the hard keys in the instrument and verify that the corresponding image of the key on the screen is highlighted, the associated LED is turned on, and an audio click is generated.

Description

1. Store the contents of the front panel display RAM in the **Front Panel Controller U103 <3>**.
 - Write *70hex* to the Command Register and read eight bytes from the Data Register.
2. Turn all the hard key LEDs off by writing *00hex* to the eight bytes in the display RAM.
3. Initialize the **Front Panel Controller U103 <3>**.
 - Write *04hex* and *22hex* to Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <3> is low.
4. Enable the front panel interrupt and the MMU interrupts. Display the image of the hard keys.
5. Wait for a hard key input.
6. When a valid hard key input is obtained, generate visual and audio feedback.
 - Highlight the image of that hard key on the screen.
 - Turn the LED associated with that hard key, if any, on by changing the contents of the display RAM in the **Front Panel Controller**.
 - Generate a click through the **Tone Generator <26>**.
7. Repeat steps 5 and 6 until an "Exit" input is received.
8. Initialize **Front Panel Controller U103 <3>**.
 - Write *04hex* and *22hex* to Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <3> is low.
9. Restore the front panel display RAM by writing back the saved contents of the display RAM.

Front Panel

Verify

Hard Keys ()

- Write *90hex* to the Command Register and write eight bytes to the Data Register.

Error Index

None.

Executive

Routine Name Soft Keys

Overview This test verifies the soft keys by allowing the user to interactively touch any of the soft keys. Both visual and audio feedback is provided. It is also useful in the calibration and verification of Crt Driver board adjustments.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off." Once this test is invoked, the operator may touch any of the soft keys in the instrument and verify that a touch box is drawn around the soft key on the screen and an audio click is generated.

Description

1. Store the contents of the front panel display RAM in the **Front Panel Controller U103 <3>**.
 - Write *70hex* to the Command Register and read eight bytes from the Data Register.
2. Initialize **Front Panel Controller U103 <3>**.
 - Write *04hex* and *22hex* to the Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <3> is low.
3. Enable the front panel interrupt and the MMU interrupts. Display a grid pattern (in dim color) along the soft key boundaries on the screen to identify them.
4. Wait for a touch input.
5. When a valid touch input is obtained, draw a box of bright lines around the selected soft key and generate a click.
6. Repeat steps 4 and 5 until an "Exit" input is received.
7. Initialize Front Panel Controller U103 <3>.
 - Write *04hex* and *22hex* to Command Register and then write *E0hex* 33 times, again to the Command Register, to assure that IRQ U103-4 <3> is low.

Error Index None.

See Also The adjustment procedure for the CRT Driver in the Service Reference Manual.

Routine Name	Knobs
Overview	This test verifies the knobs by allowing the user to interactively turn either of the knobs. Visual feedback is provided for both knob movements.
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off." Once this test is invoked, the operator may turn either of the knobs on the instrument and verify that the corresponding knob pointer on the screen rotates and its associated counter value changes.
Description	<ol style="list-style-type: none">1. Initialize the knob circuits in the Serial Data Interface U330 <25> by writing <i>80hex</i>, <i>02hex</i>, and <i>00hex</i> to Misc. Control Select 1 Register, Misc. Control Select Register, and Misc. Control Select Register, respectively.2. Enable the front panel interrupt and the MMU interrupts. Draw an image of the knobs on the screen. The display at the center of each knob image is the corresponding current knob counter value.3. Wait for either of the knobs to be turned by reading Knob Read Enable 0 and Knob Read Enable 1 Registers in the Serial Data Interface <25> repeatedly and checking for any change in their values.4. If either of the knobs is turned (i.e., the register values were different from the previous values), update the corresponding knob pointer on the screen image and update the counter value for the counter.5. Repeat steps 3 and 4 until an "Exit" input is received.
Error Index	None.

Input/Output

Temp Sensor

Comparator (E411X)

Routine Name

Comparator

Overview

This test verifies the sensitivity of the **Temperature Sensor <26>** by checking that its comparator U122 trips only once over the full temperature range.

Description

1. Exercise the comparator through the entire temperature range (0-100°C) and verify that it trips only once.
 - Write *00hex* to the **DAC Data Latch U610** (and therefore the **Temp/Tone DAC U212**) <26> and check if SET HI(H) U120C-8 is high by reading the **Temp/Tone/Diagn Readback Buffer U612** <26>. If SET HI(H) is high, then the temperature corresponding to *00hex* (0°C) is the temperature read by the sensor.
 - Increment the previous value by *06hex* (3°C increment), write it to **DAC Data Latch U610** <26>, and check if SET HI(H) U120C-8 is high. Repeat the sequence of increment/write/verify until *FFhex* (100°C) is reached. If the comparator trips only once, then SET HI(H) should be high only once for the entire temperature range.

Error Index E4111

The **Temperature Sensor <26>** comparator tripped zero times or more than one time, indicating a sensitivity problem.

See Also

The Ramp Tone manual test routine in the Tone Gen area.

Routine Name Counting

Overview This test verifies that the Real Time Clock U614 <25> is running by watching the change in its internal device counters. This test does not alter or stop the clock to perform this verification.

Description

1. Read the Hundredth Counter Register and Seconds Counter Register from the Real Time Clock U614 <25>.
2. Perform a software delay.
3. Read the Hundredth Counter Register and Seconds Counter Register again and verify that they are different from the previous readings.

Error Index E4211 The Real Time Clock <25> is not running.

Routine Name Interrupt

Overview This test verifies the Real Time Clock <25> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Mask all the Real Time Clock's <25> interrupts by writing 00hex to its Mask Register.
2. Read the Interrupt Status Register of the Real Time Clock <25> to clear all the interrupts.
3. Verify that there is no Real Time Clock interrupt at the Interrupt Controllers <28>.
 - Read the Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER and verify that IR6 U370-24 is low.
4. If there is no pending Real Time Clock interrupt at the Interrupt Controllers (i.e., IR6 U370-24 is low), enable the interrupt at the Interrupt Controllers <28> by setting Interrupt Controllers SLAVE 3 U370 <28> Mask Register bit 6 and Interrupt Controllers MASTER U350 <28> Mask Register bit 3 low.
5. Enable the Real Time Clock's one hundredth interrupt at the Real Time Clock <25> by writing 18hex and 02hex to the Command Register and Mask Register.
6. Perform a software delay (at least 10 ms).
7. Disable the Real Time Clock's one-hundredth interrupt at the Real Time Clock <25> by writing 08hex to the Command Register.
8. Verify that the Real Time Clock's interrupt did occur.
9. Disable Real Time Clock's interrupt at the Interrupt Controllers by setting Interrupt Controllers SLAVE 3 U370 <28> bit 6 and Interrupt Controllers MASTER U350 <28> bit 3 high.

Error Index E4221 The Real Time Clock interrupt did not occur.

Error Index E4222 The Real Time Clock interrupt could not be reset at the Interrupt Controllers Slave U370 <28>.

Routine Name Calibrate

Overview This test helps the operator to adjust the frequency of the **Real Time Clock** oscillator <25> to the required accuracy by generating interrupts every second.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off." After invoking this test, the operator may adjust the oscillator frequency by adjusting C510 <25>. Using a timer-counter (with an accuracy of at least ± 1 PPM) in period mode with its probe connected to TP310 <25>, C510 <25> is adjusted so that the period between two consecutive falling edges is precisely 1 s ± 5 PPM (± 5 μ s).

Description

1. Enable the front panel, and MMU interrupts, and display the "Exit" selector on the screen.
2. Enable the **Real Time Clock's** <25> interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers SLAVE 3 U370** <28> **Mask Register bit 6** and **Interrupt Controllers MASTER U350** <28> **Mask Register bit 3** low.
3. Enable **Real Time Clock's** <25> interrupt at the **Real Time Clock** <25> by writing *18hex* to the **Command Register**.
4. Until the "Exit" selection is made, repeatedly enable the **Real Time Clock's** <25> one second interrupt by writing *08hex* to the **Mask Register**.
5. Mask the **Real Time Clock's** interrupt at the **Real Time Clock** by writing *00hex* and *08hex* to the **Mask Register** and **Command Register**.
6. Disable **Real Time Clock** interrupt at the **Interrupt Controllers** by setting **Interrupt Controllers SLAVE 3 U370** <28> **Mask Register bit 6** and **Interrupt Controllers MASTER U350** <28> **Mask Register bit 3** high.

Error Index None.

See Also The adjustment procedure for the **Real Time Clock** oscillator in the **Service Reference Manual**.

Input/Output

Tone Gen

Ramp Tone ()

Routine Name Ramp Tone

Overview This test helps verify the Temp/Tone DAC U212 <26> by generating a ramp signal on the DAC output.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." After invoking this test in "cycle" mode (i.e., "LOOP" and "TERSE" selected), the operator is required to use external equipment to verify that the Temp/Tone DAC IOUT(L) U212-2 <26> is a ramp waveform with 1mA amplitude or the voltage across Tone Generator R214 <26> is a ramp waveform with 8.8V $\pm 10\%$ amplitude.

Description

1. Generate a ramp on the Temp/Tone DAC tone output.
 - Write 100_{hex} (value corresponding to the highest tone) to DAC Data Latch U610 <26>.
 - Increment the value and write it to the DAC Data Latch U610 <26>. Repeat the increment/write sequence until 1FF_{hex} is reached.
2. Turn tone generation off by writing 00_{hex} to DAC Data Latch U610 <26>.

Error Index None.

Input/Output	Printer	Loopback (E441X)
--------------	---------	------------------

Routine Name	Loopback
---------------------	----------

Overview	This test verifies the Printer Controller U430 <6>, Printer Data Buffer U520 <6>, Printer Loop Back Buffer U540 <6>, and Printer Interface Control <6> by looping back a set of patterns.
-----------------	---

Description	<ol style="list-style-type: none">1. Program the Printer Controller <6> in test mode (loop back mode) by writing <i>A6hex</i> and <i>08hex</i> to the Control Word Register so that NOR(H)/TEST(L) U430-13 is low. This enables Printer Loop Back Buffer U540 <6> and disables Printer Interface Control buffer U541 <6>.2. Transmit, receive, and verify a set of patterns.<ul style="list-style-type: none">• Write patterns <i>AAhex</i>, <i>CChex</i>, <i>F0hex</i>, <i>0Fhex</i>, and <i>55hex</i> to U430 Port A. Verify the patterns by reading U430 Port B.3. Program the Printer Controller <6> in normal mode by writing <i>09hex</i> to the Control Word Register so that NOR(H)/TEST(L) U430-13 is high.
--------------------	--

Error Index E4411	The received pattern was not the same as the transmitted one. The display shows the first pattern that did not verify.
--------------------------	--

Routine Name Interrupt

Overview This test verifies the **Printer Controller U430 <6>** interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. The interrupt is generated by looping back a byte of data.

Description

1. Program the **Printer Controller <6>** in test mode (loop back mode) by writing *A6hex* and *08hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is low. This enables **Printer Loop Back Buffer U540 <6>** and disables **Printer Interface Control buffer U541 <6>**.
2. Verify that there is no printer interrupt pending at the **Interrupt Controllers <28>**.
 - Read the **Interrupt Controllers MASTER U350 <28> INTERRUPT REQUEST REGISTER** and verify that IR6 U350-24 is low.
3. If there are no pending interrupts, enable the interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register** bit 6 low.
4. Enable the interrupt at the **Printer Controller** by writing *0Dhex* to the Control Word Register.
5. Loopback a pattern.
 - Write *AAhex* to U430 Port A, perform a software delay, and read it from U430 Port B. Immediately following the read, a transmitter empty interrupt is generated, i.e., INTR(A)(H) U430-17 is high.
6. Perform a software delay.
7. Disable the interrupt at the **Printer Controller <6>** by writing *0Chex* to the Control Word Register.
8. Disable the interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register** bit 6 high.
9. Verify that a printer interrupt did occur.
10. Program the **Printer Controller <6>** in normal mode by writing *09hex* to the Control Word Register so that NOR(H)/TEST(L) U430-13 is high.

Error Index E4421 The **Printer Controller <6>** interrupt did not occur.

Error Index E4422 The **Printer Controller <6>** interrupt could not be cleared at the **Interrupt Controllers MASTER U350 <28>**.

Routine Name	Pattern
Overview	This test helps the operator verify the Printer Controller's U430 <6> printer interface by printing a set of patterns (all printable ASCII characters).
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Before executing this test, the operator should connect a Centronics-compatible printer to the rear panel of the instrument.
Description	<ol style="list-style-type: none"> 1. Program the Printer Controller U430 <6> in normal mode with its interrupt enabled by writing <i>A6hex</i>, <i>09hex</i> and <i>0Dhex</i> to the Control Word Register. 2. Check printer availability by repeatedly reading the printer status until the printer is available or the software timeout period has expired. <ul style="list-style-type: none"> • Write <i>0Bhex</i> and <i>0Ahex</i> to the Control Word Register. • Read U430 Port B and verify that Printer Interface Control PB4(L) U541-9 <6> and PB5(H) U541-7 are high. • Read U430 Port C and verify that bit 3 is high. 3. Initialize the printer by writing <i>1Bhex</i>, <i>40hex</i>, <i>1Bhex</i>, and <i>50hex</i> to U430 Port A regardless of the printer availability. 4. Print the first line of the first character set. <ul style="list-style-type: none"> • Read the printer status using the procedure outlined in step 2. Irrespective of the printer availability, write the first pattern <i>20hex</i> (space character) to U430 Port A. Increment the pattern and write it to U430 Port A after checking for the printer availability. Repeat the sequence of increment/read printer status/write pattern to U430 Port A until 32 patterns have been output. • Write <i>0Ahex</i> (line feed) and <i>0Dhex</i> (carriage return) to U430 Port A. 5. Continue printing lines of the first character set using the procedure outlined in step 4 until the last pattern <i>7Fhex</i> (tilda character) is printed. 6. Check printer availability using procedures outlined in step 2. Irrespective of the printer availability, start a new line for second character set by writing <i>0Ahex</i> (line feed) and <i>0Dhex</i> (carriage return) to U430 Port A. 7. Print the second character set. <ul style="list-style-type: none"> • Using procedures outlined in step 4 and 5 print patterns starting from <i>A0hex</i> to <i>FFhex</i>.

Executive

Input/Output

Printer

Pattern ()

8. Check printer availability using procedures outlined in step 2. Irrespective of the printer availability, start a new line by writing *0A_{hex}* (line feed) and *0D_{hex}* (carriage return) to U430 Port A.

Error Index

None.

Caveats

This test will take longer to complete if a printer is not connected to the instrument.

Executive

Routine Name Loopback

Overview This test verifies the Std RS-232 Controller U311 <5> by internally looping back a set of patterns.

Description

1. Program Std RS-232 Controller U311 <5> in local loopback mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
 - Write *35hex*, *25hex*, *1Ahex*, *13hex*, *87hex*, *BBhex*, and *E5hex* to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
 - Read IPCR, write *00hex* to IMR, and read ISR.
 - Write *01hex*, *04hex*, *00hex*, *06hex*, *30hex*, *20hex*, and *05hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
3. Irrespective of the transmitter availability, transmit pattern *AAhex* by writing it to THRA.
4. Wait for the receiver to be ready by repeatedly reading SRA until bit 0 is high or until a software timeout period has expired.
5. Irrespective of the receiver availability, receive the pattern by reading RHRA and verify it.
6. Repeat steps 2-5 for patterns *CChex*, *F0hex*, *0Fhex*, and *55hex*.
7. Program the Std RS-232 Controller U311 <5> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
 - Write *1Ahex*, *13hex*, *07hex*, *BBhex*, and *E5hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
 - Read IPCR, write *00hex* to IMR, and read ISR.
 - Write *01hex*, *04hex*, *00hex*, *06hex*, *30hex*, *20hex*, and *05hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4511 The looped back pattern did not match the pattern sent. The display shows the first pattern that failed.

Routine Name Baud Rate

Overview

This test verifies the baud rate generator of Std RS-232 Controller U311 <5> and RS-232 Clock Generator <5> by using the Timer U822 COUNTER 2 <25> to measure the time taken to transmit and receive a character at 9600 baud.

Description

1. Program Std RS-232 Controller U311 <5> in local loopback mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
 - Write 35hex, 25hex, 1Ahex, 13hex, 87hex, BBhex, and E5hex to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
 - Read IPCR, write 00hex to IMR, and read ISR.
 - Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Enable the RS232 interrupt at the Interrupt Controllers <28> by setting Interrupt Controllers MASTER U350 <28> Mask Register bit 3 and Interrupt Controllers SLAVE 3 U370 <28> Mask Register bit 4 low.
3. Enable the receiver ready interrupt at the Std RS232 Controller U311 <5> by writing 02hex to IMR.
4. Set the baud rate of the Std RS-232 Controller <5> to 9600.
 - Write 0Ahex, BBhex, E0hex, and 05hex to CRA, CSRA, ACR, and CRA.
5. Program the Timer COUNTER 2 U822 <25> to count 6 MHz clock.
 - Write B0hex, FFhex, FFhex, and 00hex, to Timer Control Word Register, Count Register low and high bytes, and Timer Configuration Logic U720, respectively.
6. Transmit a character by writing the pattern 55hex to THRA of U311.
7. Perform a software delay.
8. If an RS-232 interrupt occurs, disable the Timer COUNTER 2 from counting by writing 40hex to Timer Configuration Logic U720 and then receive the character by reading RHRA of U311.
9. Verify the Timer COUNTER 2 U822 <25> Count Register against the known value.
10. Disable the RS232 interrupt at the Interrupt Controllers U311 <28> by setting Interrupt Controllers MASTER U350 <28> Mask Register bit 3 and Interrupt Controllers SLAVE 3 U370 <28> Mask Register bit 4 high.
11. Disable all the RS-232 interrupts at the Std RS-232 Controller <5> by writing 00hex to IMR.

Executive

12. Reset the RS-232 receiver by writing *25hex* to CRA.
13. Program the **Std RS-232 Controller U311** <5> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
 - Write *1Ahex*, *13hex*, *07hex*, *BBhex*, and *E5hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
 - Read IPCR, write *00hex* to IMR, and read ISR.
 - Write *01hex*, *04hex*, *00hex*, *06hex*, *30hex*, *20hex*, and *05hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4521

The time taken to transmit and receive a character for a baud rate was not within the expected tolerance.

Caveats

This test may fail if the Main Processor Board Clock Generator <27> reference Y950 is out of tolerance since it is being used as the timing reference for the **Timer** <25> COUNTER 2.

Routine Name Error Gen (Error Generation)

Overview This test verifies the error generation capability of the Std RS-232 Controller U311 <5> by generating an overrun error.

Description

1. Program Std RS-232 Controller U311 <5> in local loop back mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
 - Write 35hex, 25hex, 1Ahex, 13hex, 87hex, BBhex, and E5hex to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
 - Read IPCR, write 00hex to IMR, and read ISR.
 - Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
3. Irrespective of the transmitter availability, transmit pattern AAhex by writing it to THRA.
4. Repeat steps 2 and 3 for patterns CChex, F0hex, 0Fhex, 55hex, and 56hex.
5. Read SRA and verify that the overrun bit is set.
6. Reset the RS232 receiver and the error status.
 - Read RHRA twice to remove the extra characters sent.
 - Write 25hex and 45hex to CRA.
7. Program the Std RS-232 Controller U311 <5> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
 - Write 1Ahex, 13hex, 07hex, BBhex, and E5hex to CRA, MR1A, MR2A, CSR, and ACR, respectively.
 - Read IPCR, write 00hex to IMR, and read ISR.
 - Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4531 The Std RS-232 Controller <5> did not record the overrun error.

Routine Name Interrupt

Overview This test verifies the Std RS-232 Controller's interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program Std RS-232 Controller U311 <5> in local loop back mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.
 - Write 35hex, 25hex, 1Ahex, 13hex, 87hex, BBhex, and E5hex to CRA, CRA, CRA, MR1A, MR2A, CSRA, and ACR registers, respectively.
 - Read IPCR, write 00hex to IMR, and read ISR.
 - Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
2. Verify that there is no pending RS232 interrupt at the **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER** and verify that IR4 U350-22 is low.
3. If there is no pending RS232 interrupt, enable the RS-232 interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 3** and **Interrupt Controllers SLAVE 3 U370 Mask Register bit 4** low.
4. Enable the receiver ready interrupt at the Std RS-232 Controller U311 <5> by writing 02hex to IMR.
5. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired.
6. Irrespective of the transmitter availability, transmit a character by writing 55hex to THRA.
7. Perform a software delay.
8. Read the character by reading RHRA.
9. Verify that the interrupt did occur.
10. Reset the receiver and disable the receiver interrupt by writing 25hex and 00hex to CRA and IMR, respectively.
11. Disable the RS232 interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 3** and **Interrupt Controllers SLAVE 3 U370 <28> Mask Register bit 4** high.
12. Program the Std RS-232 Controller U311 <5> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate.

Executive

Input/Output

RS232

Interrupt (E454X)

- Write *1Ahex*, *13hex*, *07hex*, *BBhex*, and *E5hex* to CRA, MR1A, MR2A, CSR, and ACR, respectively.
- Read IPCR, write *00hex* to IMR, and read ISR.
- Write *01hex*, *04hex*, *00hex*, *06hex*, *30hex*, *20hex*, and *05hex* to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.

Error Index E4541

The Std RS-232 Controller <5> interrupt did not occur.

Error Index E4542

The Std RS-232 Controller <5> interrupt could not be reset at the Interrupt Controllers SLAVE U370 <28>.

Executive

Routine Name	Extern Loop (External Loopback)
Overview	This test verifies the Std RS-232 Controller U311 <5>, RS-232 Clock Generator <5> and Std RS-232 Interface Drivers <5> by externally looping back a set of patterns.
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Before running this test, the operator should connect an external loopback connector (Tek part no. 013-0198-00) to the RS232 connector on the rear panel of the instrument.
Description	<ol style="list-style-type: none"> Program the Std RS232 controller U311 <5> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate. <ul style="list-style-type: none"> Write 1Ahex, 13hex, 07hex, BBhex, and E5hex to CRA, MR1A, MR2A, CSR, and ACR, respectively. Read IPCR, write 00hex to IMR, and read ISR. Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively. Wait for the transmitter to be ready by repeatedly reading SRA until bits 2 and 3 are high or until a software timeout period has expired. Irrespective of the transmitter availability, transmit pattern AAhex by writing it to THRA. Wait for the receiver to be ready by repeatedly reading SRA until bit 0 is high or until a software timeout period has expired. Irrespective of the receiver availability, receive the pattern by reading RHRA and verify it. Repeat steps 2-5 for patterns CChex, F0hex, 0Fhex, and 55hex. Program the Std RS-232 Controller U311 <5> in normal mode with 8 bits per character, 1 stop bit, no parity and 9600 baud rate. <ul style="list-style-type: none"> Write 1Ahex, 13hex, 07hex, BBhex, and E5hex to CRA, MR1A, MR2A, CSR, and ACR, respectively. Read IPCR, write 00hex to IMR, and read ISR. Write 01hex, 04hex, 00hex, 06hex, 30hex, 20hex, and 05hex to OPBCR, OPCR, CTUR, CTLR, CRA, CRA, and CRA registers, respectively.
Error Index E4551	The pattern looped back did not match the transmitted one. The display shows the first pattern that did not verify.
Caveats	The test will fail if it is run without the external connector.

Executive

Routine Name	Intrpt Reset (Interrupt Reset)
Overview	This test verifies that the GPIB interrupt can be cleared by resetting the GPIB Controller U410 <5> .
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.
Description	<ol style="list-style-type: none">1. Reset GPIB Controller U410 <5> and verify its reset status.<ul style="list-style-type: none">• Write <i>80hex</i>, <i>00hex</i>, <i>00hex</i>, and <i>00hex</i> to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.• Read Interrupt Status Register 0 and Interrupt Status Register 1 and verify that they both read <i>00hex</i>.2. Verify that there is no GPIB interrupt pending at the Interrupt Controllers <28>.<ul style="list-style-type: none">• Read Interrupt Controllers SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER and verify that IR2 U370-20 is low.
Error Index E4611	The GPIB Controller <5> reset did not set the Interrupt Status Register 0 to <i>00hex</i> .
Error Index E4612	The GPIB Controller <5> reset did not set the Interrupt Status Register 1 to <i>00hex</i> .
Error Index E4613	The GPIB Controller <5> interrupt could not be reset at the Interrupt Controllers SLAVE 3 U370 <28> .
Caveats	This test will fail if it is run with the instrument connected to a GPIB bus.

Routine Name	Reset Status
Overview	This test verifies that the GPIB Controller U410 <5> status after a reset is correct by reading its Address Status Register and Bus Status Register.
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.
Description	<ol style="list-style-type: none">1. Reset GPIB Controller U410 <5> and verify that the Address Status Register and Bus Status Register reflect an idle state.<ul style="list-style-type: none">• Write <i>80hex</i>, <i>00hex</i>, <i>00hex</i>, and <i>00hex</i> to the Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.• Read the Address Status Register 0 and the Bus Status Register 1 and verify that they both read <i>00hex</i>.
Error Index E4621	The GPIB Controller's U410 <5> Address Status Register was not set to <i>00hex</i> after reset.
Error Index E4622	The GPIB Controller's U410 <5> Bus Status Register was not set to <i>00hex</i> after reset.
Caveats	This test will fail if it is run with the instrument connected to a GPIB bus.

Routine Name	Data Lines
Overview	This test verifies the data bus independence of the GPIB Controller U410 <5> by doing a "walking one's" test on the Data Out Register.
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.
Description	<ol style="list-style-type: none">1. Reset GPIB Controller U410 <5>.<ul style="list-style-type: none">• Write <i>80hex</i>, <i>00hex</i>, <i>00hex</i>, and <i>00hex</i> to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.2. Put the GPIB Controller U410 <5> into a "talk only" mode by writing <i>8Ahex</i> to Auxiliary Command Register.3. Verify the data line independence of the GPIB Controller U410 <5>.<ul style="list-style-type: none">• Write patterns <i>01hex</i>, <i>02hex</i>, <i>04hex</i>, <i>08hex</i>, <i>10hex</i>, <i>20hex</i>, <i>40hex</i>, and <i>80hex</i> to Data Out Register, read it back from Command Pass Through Register, and verify them.4. Unset the GPIB Controller's U410 <5> "talk only" mode by writing <i>0Ahex</i> to Auxiliary Command Register.
Error Index E4631	The read pattern did not match the written pattern. The display shows the first pattern that did not verify.
Caveats	This test will fail if it is run with the instrument connected to a GPIB bus.

Input/Output

GPIB

Interrupt (E464X)

Routine Name Interrupt

Overview This test verifies the **GPIB Controller's** U410 <5> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated. A "buffer empty" interrupt is used for verification.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off." Disconnect the instrument from the GPIB bus before running this test.

Description

1. Reset **GPIB Controller** U410 <5>.
 - Write *80hex*, *00hex*, *00hex*, and *00hex* to Auxiliary Command Register, Interrupt Mask Register 0, Interrupt Mask Register 1, and Auxiliary Command Register, respectively.
2. Verify that there is no GPIB interrupt pending at the **Interrupt Controllers** <28>.
 - Read **Interrupt Controllers** SLAVE 3 U370 <28> INTERRUPT REQUEST REGISTER and verify that IR2 U370-20 is low.
3. If there is no pending GPIB interrupt, enable GPIB interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <28> Mask Register bit 2 low.
4. Enable the "byte out" interrupt at the GPIB Controller by writing *10hex* to the Interrupt Mask Register 0.
5. Put the **GPIB Controller** in "talk only" mode by writing *8Ahex* to the Auxiliary Command Register.
6. Perform a software delay.
7. Disable the "byte out" interrupt at the **GPIB Controller** by writing *00hex* to the Interrupt Mask Register 0.
8. Unset the **GPIB Controller's** "talk only" mode by writing *0Ahex* to Auxiliary Command Register.
9. Disable the interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 3 and **Interrupt Controllers** SLAVE 3 U370 <28> Mask Register bit 2 high.
10. Verify that the GPIB interrupt did occur.

Error Index E4641 The **GPIB Controller** <5> interrupt did not occur.

Error Index E4642 The **GPIB Controller** <5> interrupt could not be reset at the **Interrupt Controllers** SLAVE 3 U370 <28>.

Executive

Input/Output

GPIB

Interrupt (E464X)

Caveats

This test will fail if it is run with the instrument connected to a GPIB bus.

Executive

Routine Name Status Reg (Status Register)

Overview This test verifies MMU Gate Array's U210 <33> Status and Mode Register (SMR) by performing a pattern test on the register.

Description

1. Reset the MMU Gate Array U210 <33> to diagnostic mode by writing *1Ahex*, *007Fhex*, and *0hex* to **Diagnostics U524 <33>**, **SMR**, and **Diagnostics U530 <33>**, respectively.
2. Perform a pattern test on SMR on bits that are both readable and writable.
 - Write, read, and verify patterns *0001hex*, *0002hex*, *0020hex*, *0040hex*, *0080hex* and *0000hex*.
3. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to **Diagnostics U530 <33>** and U210 Status and Mode Register.

Error Index E5111 The pattern read did not match the pattern written. The display shows the first pattern that failed.

Routine Name Arbitration

Overview

This test verifies the arbitration capability of the MMU Gate Array U210 <33> (hereafter referred to as MMU). With the MMU in single-step mode, each arbitration combination is carefully set up and the MMU is single-stepped to the completion of that arbitration cycle. The outcome of the arbitration is verified against known results.

Description

1. Reset the MMU to diagnostic mode by writing *1Ahex*, *007Fhex*, and *0hex* to **Diagnostics U524 <33>**, **U210 Status and Mode Register (SMR)**, and **Diagnostics U530 <33>**.
2. Find a "refresh" arbitration cycle.
 - Pulse Clock Generator CLK(H) U110D-11 <33> (by toggling **Diagnostics U530-7**) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.
3. Irrespective of finding a "refresh" cycle, either remove the "refresh" cycle or find the next "refresh" cycle.
 - If the current arbitration group does not have "refresh" cycle in it, then pulse CLK(H) U110D-11 20 times to complete the "refresh" cycle. Else, pulse the clock 252 times to find the next "refresh" cycle.
4. Generate "read digitizer" and "read display" arbitration requests at their sync positions.

"Read Digitizer"

- Write *0000hex* and *00CChex* to the Random Address Generator (RAG) and RAG Register 0. Set bit 7 of SMR to set the MMU Gate Array to diagnostic mode so that the tag bits are assured to be *0000hex*.
- Start the "read digitizer" arbitration by setting DIGREQ(L) U524-12 low and then high.
- Pulse CLK(H) U110D-11 <33> appropriate number of times (six times for this combination) so that the all the other arbitration requests of this arbitration combination will coincide at the sync position.

"Read Display"

- Write *0000hex*, *00A9hex* and *00AAhex* to SAG Message Pointer Register (MPR), Message Length Register (MLR) and Address Counter (AC) to set up the SAG to "read display." Set bit 10 of SMR to start the "read display" channel.

Executive

- Start the "read display" arbitration by setting DATARDY(H) U524-6 high and then low.
 - Pulse CLK(H) U110D-11 <33> appropriate number of times (1) so that all the other arbitration requests of this arbitration combination will coincide at their sync positions.
5. Pulse CLK(H) U110D-11 <33> appropriate number of times to read the MMU's output at Diagnostics U112 <33>. The low and high bytes are read separately from Diagnostics U112 <33>. For each arbitration request, the output is verified against a known value.
 6. Complete MMU arbitration requests by pulsing CLK(H) U110D-11 <33> to the corresponding completion of each request.
 7. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register.
 8. Repeat steps 2-7 for the following arbitration combinations:
 - case 2: "write display" and "read digitizer"
 - case 3: "refresh" and "read display"
 - case 4: "refresh" and "read digitizer"
 - case 5: "refresh", "read digitizer", and "read display"
 - case 6: "refresh" and "write display"
 - case 7: "refresh", "write display", and "read digitizer"
 - case 8: "refresh" and "write digitizer"

The other arbitration requests are set up in the following ways:

"Refresh"

Since the "refresh" cycle occurs automatically every 256 clock cycles, it is only necessary to find the start of a "refresh" cycle and set up the other arbitration requests to coincide with the next "refresh" cycle's sync point.

"Write Display"

- Write *0000hex*, *00A9hex* and *00AAhex* to SAG MPR, MLR and AC to set up the SAG to "write display." Set bit 9 of SMR to start the "write display" channel.
- Start "write display" arbitration by setting SENDNEW(L) U524-5 low and then high.
- Pulse CLK(H) U110D-11 <33> appropriate number of times so that all the other arbitration requests of this arbitration combination will coincide at the sync position.

"Write Digitizer"

Executive

- Write 0000_{hex}, 00A9_{hex} and 00AA_{hex} to SAG MPR, MLR and AC to set up the SAG to "write digitizer". Set bit 11 of SMR to start "write digitizer" channel.
- Start the "write digitizer" arbitration by setting DIGACK(L) U524-9 low and then high.
- Pulse CLK(H) U110D-11 <33> appropriate number of times so that all the other arbitration requests of this arbitration combination will coincide at the sync position.

Error Index E5121	Combination "read digitizer" and "read display" failed.
Error Index E5122	Combination "write display" and "read digitizer" failed.
Error Index E5123	Combination "refresh" and "read display" failed.
Error Index E5124	Combination "refresh" and "read digitizer" failed.
Error Index E5125	Combination "refresh", "read digitizer", and "read display" failed.
Error Index E5126	Combination "refresh" and "write display" failed.
Error Index E5127	Combination "refresh", "write display", and "read digitizer" failed.
Error Index E5128	Combination "refresh" and "write digitizer" failed.

Routine Name Refresh

Overview This test verifies the MMU Gate Array U210 <33> Refresh Counter by performing a "walking one's" test on this register.

Description

1. Reset the MMU U210 <33> to diagnostic mode by writing `1Ahex`, `007Fhex`, and `0hex` to **Diagnostics U524 <33>**, **U210 Status and Mode Register (SMR)**, and **Diagnostics U530 <33>**.
2. Find a "refresh" arbitration cycle.
 - Pulse CLK(H) U110D-11 <33> (by toggling **Diagnostics U530-7**) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU.
3. Process the "refresh" cycle by pulsing the clock 20 times.
4. Make the Refresh Counter readable at the **Diagnostics U112 <33>**. The Refresh Counter is readable only when the refresh cycle is at its sync point of arbitration.
 - Pulse CLK(H) U110D-11 <33> until the next refresh cycle (255 clocks from the previous "refresh" cycle) is found. A maximum of five attempts will be made to find two consecutive "refresh" cycles that are 255 clocks apart.
5. Irrespective of finding two consecutive "refresh" cycles, perform a "walking one's" pattern test on the Refresh Counter.
 - Write patterns `0001hex`, `0002hex`, `0004hex`, `0008hex`, `0010hex`, `0020hex`, `0040hex`, and `0080hex` to the Refresh Counter. Read and verify the patterns from **Diagnostics U112 <33>**.
6. Reset the MMU Gate Array U210 <33> to normal mode by writing `5hex` and `007Fhex` to **Diagnostics U530 <33>** and **U210 Status and Mode Register**.

Error Index E5131 The Refresh Counter of U210 could not be accessed.

Error Index E5132 The pattern read from the Refresh Counter did not match the pattern written. The display shows the first pattern that failed.

Routine Name Dtalk Intrpt (Display Talk Request Interrupt)

Overview This test verifies the MMU Gate Array U210 <33> display talk request interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the MMU Gate Array U210 <33> to an isolated diagnostic mode by writing 1Ahex, 007Fhex, and 4hex to **Diagnostics U524 <33>**, U210 Status and Mode Register (SMR), and **Diagnostics U530 <33>**, respectively.
2. Verify that there is no pending display talk request interrupt at the **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER** and verify that IR2 U360-20 is low.
3. If there is no display interrupt pending, then enable display talk request interrupt by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 2** low.
4. Generate a display talk request interrupt by setting DATARDY(H) U524-6 <33> high and then low.
5. Perform a software delay and then verify that the display talk request interrupt did occur.
6. Clear the interrupt at the MMU by setting SMR bit 1 high.
7. Disable the display talk request interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 2** high.
8. Reset the MMU Gate Array U210 <33> to normal mode by writing 5hex and 007Fhex to **Diagnostics U530 <33>** and U210 Status and Mode Register.

Error Index E5141 The display talk request interrupt did not occur.

Error Index E5142 The display talk request interrupt could not be cleared at the **Interrupt Controllers SLAVE 1 U360 <28>**.

Routine Name SAG Compare (Sequential Address Generator Comparator)

Overview This test verifies the MMU Gate Array's U210 <33> Sequential Address Generator (SAG) comparator by comparing a fixed value in the Address Counter (AC) with different values in the Message Length Register (MLR).

Description

1. Reset the MMU Gate Array U210 <33> to diagnostic mode by writing *1Ahex*, *007Fhex*, and *0hex* to Diagnostics U524 <33>, U210 Status and Mode Register (SMR), and Diagnostics U530 <33>, respectively.
2. Program the SAG to write a word to the display.
 - Write *0000hex*, *7FFFhex* (since the SAG pre-increments the AC, this value becomes *8000hex* for each comparison), and *0001hex* to Message Pointer Register (MPR), AC, and MLR, respectively.
3. Start a "write display" cycle by setting bit 9 of SMR high and setting SENDNEW(L) U524-5 <33> low and then high.
4. Pulse Clock Generator CLK(H) U110D-11 (by toggling Diagnostics U530-7) until MDATALATCH(L) U400-5 is low.
5. Pulse CLK(H) U110D-11 6 times to complete "write display" cycle.
6. Read and verify the output of the comparator. The output of the comparator is reflected in bit 3 of SMR. If the AC and MLR are equal, bit 3 will be high. The comparator output should be high only when the value in the MLR is *8000hex*.
7. Clear SAG interrupt, if any, by setting bit 3 of SMR high.
8. Repeat steps 2-7 for values *0002hex*, *0004hex*, *0008hex*, *0010hex*, *0020hex*, *0040hex*, *0080hex*, *0100hex*, *0200hex*, *0400hex*, *0800hex*, *1000hex*, *2000hex*, *4000hex*, and *8000hex* in MLR.
9. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register.

Error Index E5151 The SAG comparator output was incorrect. The display shows the first MLR register value for which the comparator output failed.

Routine Name SAG Adder (Sequential Address Generator Adder)

Overview This test verifies the SAG adder in the MMU Gate Array U210 <33> by adding the Address Counter (AC) and the Message Pointer Register (MPR).

Description

1. Reset the MMU Gate Array U210 <33> to diagnostic mode by writing *1Ahex*, *007Fhex*, and *0hex* to Diagnostics U524 <33>, U210 Status and Mode Register (SMR), and Diagnostics U530 <33>, respectively.
2. Find a "refresh" arbitration cycle.
 - Pulse Clock Generator CLK(H) U110D-11 <33> (by toggling Diagnostics U530-7) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.
3. Process the "refresh" cycle by pulsing CLK(H) U110D-11 20 times.
4. Program the SAG to read from display by writing *0002hex*, *0001hex* and *0001hex* to MLR, AC and MPR.
5. Start a "read display" cycle by setting bit 10 in SMR and setting DATARDY(H) U524-6 <33> high and then low.
6. Find the "read display" cycle.
 - Pulse CLK(H) U110D-11 <33> until a SAG RAS is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.
7. Read the low byte of the adder output from Diagnostics U112 <33>.
8. Pulse CLK(H) U110D-11 two times.
9. Read the upper byte of the adder output from Diagnostics U112 <33>. Swap bits 9 and 15 of the upper byte.
10. Verify the adder output against the known output.
11. Complete the "read display" cycle by pulsing CLK(H) U110D-11 four times and clear the SAG interrupt, if any, by setting bit 3 of SMR.
12. Repeat steps 4 -11 for *0002hex*, *0004hex*, *0008hex*, *0010hex*, *0020hex*, *0040hex*, *0080hex*, *0100hex*, *0200hex*, *0400hex*, *0800hex*, *1000hex*, *2000hex*, *4000hex*, and *8000hex* in AC and MPR.
13. Program the SAG to read from display by writing *0002hex*, *0010hex* and *3FFFhex* to the MLR, AC and MPR.
14. Complete the "read display" cycle by following procedures outlined in steps 5-11 and verify that the adder output rippled to value *0000hex*.

Routine Name	SAG Intrpt (Sequential Address Generator Interrupt)
Overview	This test verifies the MMU Gate Array U210 <33> SAG Interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.
Description	<ol style="list-style-type: none"> 1. Reset the MMU Gate Array U210 <33> to an isolated diagnostic mode by writing 1A_{hex}, 007F_{hex}, and 4_{hex} to Diagnostics U524 <33>, U210 Status and Mode Register (SMR), and Diagnostics U530 <33>, respectively. 2. Verify that there is no pending SAG interrupt at the Interrupt Controllers <28>. <ul style="list-style-type: none"> • Read Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER and verify that IR0 U360-18 is low. 3. If there is no SAG interrupt pending, enable the SAG interrupt at the Interrupt Controllers <28> by setting Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low. 4. Start a "write display" cycle by setting bit 9 of SMR high and setting SENDNEW(L) U524-5 <33> low and then high. 5. Perform a software delay and then verify that the SAG interrupt did occur. 6. Clear SAG interrupt at the MMU by setting bit 3 of SMR high. 7. Disable the SAG interrupt at the Interrupt Controllers <28> by setting Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 high. 8. Reset the MMU Gate Array U210 <33> to normal mode by writing 5_{hex} and 007F_{hex} to Diagnostics U530 <33> and U210 Status and Mode Register.
Error Index E5171	The SAG interrupt did not occur.
Error Index E5172	The SAG interrupt could not be cleared at the Interrupt Controllers SLAVE 1 U360 <28> .

Routine Name

RAG Regs (Random Address Generator Registers)

Overview

This test verifies all the MMU Gate Array U210 <33> RAG registers by performing a "walking one's" test on each 9-bit RAG register.

Description

1. Reset the MMU Gate Array U210 <33> to diagnostic mode by writing *1Ahex*, *007Fhex*, and *0hex* to **Diagnostics U524 <33>**, U210 Status and Mode Register (SMR), and **Diagnostics U530 <33>**, respectively.
2. Find a "refresh" arbitration cycle.
 - Pulse Clock Generator CLK(H) U110D-11 <33> (by toggling **Diagnostics U530-7**) until a "refresh" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.
3. Process the "refresh" cycle by pulsing CLK(H) U110D-11 20 times.
4. Start a "read digitizer" cycle by setting DIGREQ(L) U524-12 <33> low and then high.
5. Find the "read digitizer" cycle.
 - Pulse CLK(H) U110D-11 <33> until a "read digitizer" cycle is detected in the Internal Diagnostic Status Register of the MMU or until the number of clock pulses exceeds 512.
6. Select the RAG Register 0 by writing the tag number of the register (*0000hex*) to RAG Tag Register in the MMU Gate Array.
7. Write pattern *0001hex* to RAG Register 0.
8. Start the data handshake by setting DIGACK(H) U524-9 <33> high and then low.
9. Pulse CLK(H) U110D-11 two times and read the low byte of the RAG output from **Diagnostics U112**. Pulse CLK(H) U110D-11 two more times and read the high byte of the RAG output from **Diagnostics U112 <33>**. Swap Bit 9 with bit 16 of the RAG output.
10. Verify the RAG output against the known value.
11. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to **Diagnostics U530 <33>** and U210 Status and Mode Register.
12. Repeat steps 1-11 for patterns *0002hex*, *0004hex*, *0008hex*, *0010hex*, *0020hex*, *0040hex*, *0080hex*, and *0100hex* in RAG Register 0.
13. Repeat steps 1-12 for RAG Registers 1-14.
14. Restore the RAG register 14 for normal operating firmware.

Executive

Subsys Comm

MMU Control

RAG Regs (E518X)

Error Index E5181	RAG register 0 failed the test.
Error Index E5182	RAG register 1 failed the test.
Error Index E5183	RAG register 2 failed the test.
Error Index E5184	RAG register 3 failed the test.
Error Index E5185	RAG register 4 failed the test.
Error Index E5186	RAG register 5 failed the test.
Error Index E5187	RAG register 6 failed the test.
Error Index E5188	RAG register 7 failed the test.
Error Index E5189	RAG register 8 failed the test.
Error Index E518A	RAG register 9 failed the test.
Error Index E518B	RAG register 10 failed the test.
Error Index E518C	RAG register 11 failed the test.
Error Index E518D	RAG register 12 failed the test.
Error Index E518E	RAG register 13 failed the test.
Error Index E518F	RAG register 14 failed the test.

Routine Name RAG Intrpt (Random Address Generator Interrupt)

Overview This test verifies MMU Gate Array's U210 <33> RAG interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Reset the MMU Gate Array U210 <33> to an isolated diagnostic mode by writing *1Ahex*, *007Fhex*, and *4hex* to **Diagnostics U524 <33>**, **U210 Status and Mode Register (SMR)**, and **Diagnostics U530 <33>**, respectively.
2. Verify that there is no RAG interrupt pending at **Interrupt Controllers <28>**.
 - Read **Interrupt Controllers SLAVE 1 U360 <28> INTERRUPT REQUEST REGISTER** and verify that **IR1 U360-19** is low.
3. If there is no RAG interrupt pending, enable the RAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 1** low.
4. Generate an "end of digitizer message" interrupt.
 - Set **DIGREQ(L) U524-12 <33>** low and then high with **U524-19(H):H <33>**.
 - Set **DIGREQ(L) U524-12 <33>** high with **U524-19(H):L <33>**.
5. Perform a software delay and then verify that the RAG interrupt did occur.
6. Clear the RAG interrupt at the MMU by setting bit 4 of the **Status and Mode Register (SMR)** high.
7. Disable RAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 1** high.
8. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to **Diagnostics U530 <33>** and **U210 Status and Mode Register**.

Error Index E5191 The RAG interrupt did not occur.

Error Index E5192 The RAG interrupt could not be cleared at the **Interrupt Controllers SLAVE U360 <28>**.

Routine Name	Size
--------------	------

Overview	This test verifies that each 64k byte waveform RAM memory segment (there are four of them) in Odd DRAM <34> and Even DRAM <34> can be accessed. This is done by writing and reading one test pattern to each segment.
----------	---

Description	<ol style="list-style-type: none"> 1. Reset MMU Gate Array U210 <33> to normal mode by writing <i>5hex</i> and <i>007Fhex</i> to Diagnostics U530 <33> and U210 Status and Mode Register (SMR). 2. Disable "read digitizer" channel by setting bit 6 of SMR low. 3. Write patterns <i>0001hex</i>, <i>0002hex</i>, <i>0004hex</i>, and <i>0008hex</i> to waveform RAM locations <i>40000hex</i>, <i>50000hex</i>, <i>60000hex</i>, and <i>70000hex</i>, respectively. 4. Read all the locations that were written and verify that their contents have remained the same. 5. Enable "read digitizer" channel by setting bit 6 of SMR high.
-------------	--

Error Index E5211	One of the patterns read from the first location in a 64k byte waveform memory segment was not the pattern written.
-------------------	---

Routine Name Data Lines

Overview This test verifies the data line independence of waveform RAM by performing a "walking one's" test on an Odd DRAM <34> location.

Description

1. Reset MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register (SMR).
2. Disable "read digitizer" channel by setting bit 6 of SMR low.
3. Perform a "walking one's" test on Odd DRAM location *7FFFEhex*.
 - Write, read, and verify patterns *0001hex*, *0002hex*, *0004hex*, *0008hex*, *0010hex*, *0020hex*, *0040hex*, *0080hex*, *0100hex*, *0200hex*, *0400hex*, *0800hex*, *1000hex*, *2000hex*, *4000hex*, and *8000hex*.
4. Enable "read digitizer" channel by setting bit 6 of SMR high.

Error Index E5221 The pattern read from Odd DRAM <34> memory location *7FFFEhex* was not the pattern written. The correspondence of the data bits to the Odd DRAM devices is shown in the following table:

<u>D0—D3</u>	<u>D4—D7</u>	<u>D8—D11</u>	<u>D12—D15</u>
U130	U124	U122	U120

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of Odd DRAM <34> and Even DRAM <34> by performing a RAM test on all locations.

Description

1. Reset the MMU Gate Array U210 <33> to normal mode by writing *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register (SMR).
2. Disable "read digitizer" channel by setting bit 6 of SMR low.
3. Verify waveform RAM address range *40000hex-4FFFFhex* (64k bytes). Terminate test if any verify operation fails.
 - Fill the address range *40000hex-4FFFFhex* with the pattern *AAAAhex*.
 - Read and verify address *40000hex* for *AAAAhex*. If so, write *CCCChex* to address *40000hex*. Increment address and continue this read/verify/write sequence until address *4FFFFhex* is reached.
 - Repeat the read/verify/write sequence, starting again at *40000hex* for *CCCChex*, *5555hex*, and *0000hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
4. Repeat step 3 for waveform RAM address ranges *50000-5FFFFhex*, *60000-6FFFFhex*, and *70000-7FFFFhex*.

Error Index E5231

The pattern read from an Odd DRAM <34> or Even DRAM <34> memory location did not match the pattern written. The correspondence of the address and data bits to DRAM memory devices is shown in the following table:

Address(hex)	<u>D0—D3</u>	<u>D4—D7</u>	<u>D8—D11</u>	<u>D12—D15</u>
XXXX0	U330	U324	U322	U320
XXXX4	U330	U324	U322	U320
XXXX8	U330	U324	U322	U320
XXXXC	U330	U324	U322	U320
XXXX2	U130	U124	U122	U120
XXXX6	U130	U124	U122	U120
XXXXA	U130	U124	U122	U120
XXXXE	U130	U124	U122	U120

X—Don't Care.

Routine Name Reset

Overview This test verifies the Waveform Compressor/Adder (WCA) Reset Control <35> by checking that the WCA returns to a correct state after reset.

Description 1. Reset WCA.

- Toggle Reset Control U130D-11 WCARST(H) by writing 0000_{hex} to Address Decode/Select port 1002_{hex} <35>.
- Write 01_{hex} to the Compression Factor Counter Register U824 <35>, and 05_{hex} to the Mode Select Latch U724 <35>.

2. Read WCA Status Read Back buffer U822 <35> and verify that it reads 04_{hex}.

Error Index E5311 The WCA does not return to a correct state after reset. The information read back from the Status Read Back buffer U822 <35> is shown below.

bit 0 (D0) : ADDAV(L)
bit 1 (D1) : IDLE(L)
bit 2 (D2) : OVER(H)
bit 3 (D3) : UNDER(H)
bit 4 (D4) : NULL(H)
bit 5 (D5) : M/MUND(H)
bit 6 (D6) : ALLNULL(H)
bit 7 (D7) : EOC(H)

Routine Name Allnull

Overview This test verifies the capability of **Compressor Control** <35> and **Flags Decode** <36> to recognize Null and non-Null data values. **Status Read Back** bit ALLNULL <35> is observed while sending three sets of Null and non-Null data values from the **MMU Gate Array** <33> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <28>.
 - Write *5hex* and *007Fhex* to **Diagnostics** U530 <33> and U210 Status and Mode Register.
 - Set **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 4.
 - Toggle **Reset Control** U130D-11 WCARST(H) <35> by writing *0000hex* to **Address Decode/Select** port *1002hex* <35>.
 - Write *04hex* and *05hex* to the **Compression Factor Counter Register** U824 <35> and the **Mode Select Latch** U724 <35>.
3. Manipulate the **MMU Gate Array** <33> to send a Null data value (*8000hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <35> and verify that **Compressor Control** U120B-9 ALLNULL(H) <35> is high.
6. Manipulate the **MMU Gate Array** <33> to send two words (*8200hex*, *8000hex*) from the waveform RAM to the WCA.
7. Perform a software delay.
8. Read **Status Read Back** buffer U822 <35> and verify that **Compressor Control** U120B-9 ALLNULL(H) <35> is low.
9. Manipulate the **MMU Gate Array** <33> to send two more words (*8000hex*, *8200hex*) from the waveform RAM to the WCA.
10. Perform a software delay.
11. Read **Status Read Back** buffer U822 <35> and verify that **Compressor Control** U120B-9 ALLNULL(H) <35> is low.

12. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <35>.
13. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers MASTER** U350 <28> Mask Register bit 1 and **Interrupt Controllers SLAVE 1** U360 <28> Mask Register bit 0 low.

Error Index E5321 The WCA did not recognize a Null data value, i.e., the ALLNULL bit read from **Status Read Back** buffer U822 <35> was low when it should have been high.

Error Index E5322 The WCA did not recognize a non-Null data value, i.e., the ALLNULL bit read from **Status Read Back** buffer U822 <35> was low when it should have been high.

Error Index E5323 The WCA did not recognize a non-Null data value in a group containing both Null and non-Null data values.

Routine Name M/Mund

Overview This test verifies that **Compressor Control** <35> is capable of selecting new minimum and maximum data points for each data group in the non-vectorized mode and is capable of using the previous minimum and maximum data points for the next data group in the vectored mode. **Status Read Back** bit M/MUND <35> is observed while sending two sets of two words from the **MMU Gate Array** <33> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <28>.
 - Write *5hex* and *007Fhex* to **Diagnostics** U530 <33> and U210 Status and Mode Register.
 - Set **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectorized test mode with a compression factor of 3.
 - Toggle **Reset Control** U130D-11 WCARST(H) <35> by writing *0000hex* to **Address Decode/Select** port *1002hex* <35>.
 - Write *03hex* and *05hex* to the **Compression Factor Counter** Register U824 <35> and the **Mode Select Latch** U724 <35>.
3. Manipulate the **MMU Gate Array** <33> to send two words (*8200hex* and *8300hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <35> and verify that **Compressor Control** U124B-9 M/MUND(H) <35> is high, i.e., new minimum and maximum data points need to be selected.
6. Manipulate the **MMU Gate Array** <33> to send two more words (*8400hex* and *8500hex*) from the waveform RAM to the WCA.
7. Perform a software delay.
8. Read **Status Read Back** buffer U822 <35> and verify that **Compressor Control** U124B-9 M/MUND(H) <35> is high, i.e., new minimum and maximum data points need to be selected.
9. Repeat steps 3-8 with the WCA in vectored mode by writing *07hex* to **Mode Select Latch** U724 <35> and verify that M/MUND(H) <35> is high in step 5 and low in step 8.

10. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
11. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0** low.

Error Index E5331 **Compressor Control M/MUND(H) <35>** was low when the minimum and maximum data points were undefined in non-vectorred mode.

Error Index E5332 **Compressor Control M/MUND(H) <35>** was low after one cycle in non-vectorred mode. In non-vectorred mode, M/MUND(H) should have been high after each cycle since the old minimum and maximum data points are not remembered.

Error Index E3333 **Compressor Control M/MUND(H) <35>** was low when the minimum and maximum data points were not defined under vectored mode.

Error Index E5334 **Compressor Control M/MUND(H) <35>** was high after one cycle in vectored mode. In vectored mode, the previous minimum and maximum data points are remembered. So, they will not be undefined.

Routine Name Idle

Overview This test verifies that **Compressor Control** <35> and **Adder Control** <37> reflect the correct state (busy or idle) of the compressor and the adder. **Status Read Back** buffer bits IDLE(L) and ADDAV(L) <35> are observed while sending two sets of two words from the **MMU Gate Array** <33> to the Compressor.

Description

1. Initialize the **MMU Gate Array** U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <28>.
 - Write *5hex* and *007Fhex* to **Diagnostics** U530 <33> and U210 Status and Mode Register.
 - Set **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 2.
 - Toggle **Reset Control** U130D-11 WCARST(H) by writing *0000hex* to **Address Decode/Select** port *1002hex* <35>.
 - Write *02hex* and *05hex* to the **Compression Factor Counter** Register U824 <35> and the **Mode Select Latch** U724 <35>.
3. Manipulate the **MMU Gate Array** <33> to send two words (*8200hex* and *8300hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Verify that the compressor is idle and the adder is busy.
 - Read **Status Read Back** buffer U822 <35> and verify that **Compressor Control** U222-17 IDLE(L) <35> is low and **Adder Control** U224-19 ADDAV(L) <37> is high.
6. Manipulate the **MMU Gate Array** <33> to send two more words (*8400hex* and *8500hex*) from the waveform RAM to the WCA.
7. Perform a software delay.
8. Verify that both the compressor and the adder are busy.
 - Read **Status Read Back** buffer U822 <35> and verify that **Compressor Control** U222-17 IDLE(L) <35> is high and **Adder Control** U224-19 ADDAV(L) <37> is high.
9. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <35>.

10. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers MASTER** U350 <28> Mask Register bit 1 and **Interrupt Controllers SLAVE 1** U360 <28> Mask Register bit 0 low.

Error Index E5341 The compressor was not busy when it should have been.

Error Index E5342 The adder and/or the compressor was not busy when it should have been.

Subsys Comm	WCA Control	Comprss Null (E535X)
Routine Nam	Comprss Null (Compress Null)	
Overview	This test verifies that Flags Decode <36> detects a Null data point appropriately in compress mode by sending the Compressor a Null data point (8000hex).	
Description	<ol style="list-style-type: none"> 1. Initialize the MMU Gate Array U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the Interrupt Controllers <28>. <ul style="list-style-type: none"> • Write 5hex and 007Fhex to Diagnostics U530 <33> and U210 Status and Mode Register. • Set Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low. 2. Reset WCA to non-vectored test mode with a compression factor of 1. <ul style="list-style-type: none"> • Toggle Reset Control U130D-11 WCARST(H) by writing 0000hex to Address Decode/Select port 1002hex <35>. • Write 01hex and 05hex to the Compression Factor Counter Register U824 <35> and the Mode Select Latch U724 <35>. 3. Manipulate the MMU Gate Array <33> to send a Null data point (8000hex) from the waveform RAM to the WCA. 4. Perform a software delay. 5. Read Status Read Back buffer U822 <35> and verify that Flags Decode U100C-8 NULL(H) <36> is high. 6. Reset the WCA by toggling Reset Control U130D-11 WCARST(H) <35>. 7. Disable SAG interrupt at the Interrupt Controllers <28> by setting Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low. 	
Error Index E5351	The Null data point is not being detected as a null condition by the WCA in compress mode.	

Executive

Routine Name Xparent Null (Transparent Null)

Overview This test verifies that **Flags Decode <36>** detects a Null data point appropriately in transparent mode by sending the Compressor a Null data point (8000hex).

Description

1. Initialize the **MMU Gate Array U210 <33>** and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers <28>**.
 - Write 5hex and 007Fhex to **Diagnostics U530 <33>** and U210 Status and Mode Register.
 - Set **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
 - Toggle **Reset Control U130D-11 WCARST(H)** by writing 0000hex to **Address Decode/Select port 1002hex <35>**.
 - Write 01hex and 04hex to the **Compression Factor Counter Register U824 <35>** and the **Mode Select Latch U724 <35>**.
3. Manipulate the **MMU Gate Array <33>** to send a Null data point (8000hex) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back buffer U822 <35>** and verify that **Flags Decode U100C-8 NULL(H) <36>** is low.
6. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
7. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.

Error Index E5361 The Null data value is being detected as a Null condition by the WCA in the transparent mode when it shouldn't be.

Routine Name Comprss Over (Compress Overrange)

Overview This test verifies that **Flags Decode <36>** detects an Over data point appropriately in the compress mode by sending the Compressor an overrange value (*7FFFhex*).

Description

1. Initialize the **MMU Gate Array U210 <33>** and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers <28>**.
 - Write *5hex* and *007Fhex* to **Diagnostics U530 <33>** and **U210 Status and Mode Register**.
 - Set **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 1.
 - Toggle **Reset Control U130D-11 WCARST(H)** by writing *0000hex* to **Address Decode/Select port 1002hex <35>**.
 - Write *01hex* and *05hex* to the **Compression Factor Counter Register U824 <35>** and the **Mode Select Latch U724 <35>**.
3. Manipulate the **MMU Gate Array <33>** to send an overrange data value (*7FFFhex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back buffer U822 <35>** and verify that **Flags Decode U100A-3 OVER(H) <36>** is high.
6. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
7. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.

Error Index E5371 The overrange data value is not being decoded into the OVER flag properly in the compress mode.

Routine Name Xparent Over (Transparent Overrange)

Overview This test verifies that **Flags Decode** <36> detects an Over data point appropriately in the transparent mode by sending the Compressor an overrange value (7FFF_{hex}).

Description

1. Initialize the **MMU Gate Array** U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <28>.
 - Write 5_{hex} and 007F_{hex} to **Diagnostics** U530 <33> and U210 Status and Mode Register.
 - Set **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
 - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000_{hex} to **Address Decode/Select** port 1002_{hex} <35>.
 - Write 01_{hex} and 04_{hex} to the **Compression Factor Counter** Register U824 <35> and the **Mode Select Latch** U724 <35>.
3. Manipulate the **MMU Gate Array** <33> to send an overrange data value (7FFF_{hex}) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <35> and verify that **Flags Decode** U100A-3 OVER(H) <36> is low.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <35>.
7. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.

Error Index E5381 The overrange data value is being decoded into the OVER flag in the transparent mode when it shouldn't be.

Routine Name Comprss Undr (Compress Underrange)

Overview This test verifies that **Flags Decode <36>** detects an Under data point appropriately in the compress mode by sending the Compressor an underrange data value (8001*hex*).

Description

1. Initialize the **MMU Gate Array U210 <33>** and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers <28>**.
 - Write 5*hex* and 007F*hex* to **Diagnostics U530 <33>** and **U210 Status and Mode Register**.
 - Set **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 1.
 - Toggle **Reset Control U130D-11 WCARST(H)** by writing 0000*hex* to **Address Decode/Select port 1002*hex* <35>**.
 - Write 01*hex* and 05*hex* to the **Compression Factor Counter Register U824 <35>** and the **Mode Select Latch U724 <35>**.
3. Manipulate the **MMU Gate Array <33>** to send an underrange data value (8001*hex*) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back buffer U822 <35>** and verify that **Flags Decode U100B-6 UNDER(H) <36>** is high.
6. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
7. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.

Error Index E5391 The underrange data value is not being decoded properly into the UNDER flag in the compress mode.

Routine Name Xparent Undr (Transparent Underrange)

Overview This test verifies that **Flags Decode** <36> detects an Under data point appropriately in the transparent mode by sending the Compressor an underrange data value (8001hex).

Description

1. Initialize the **MMU Gate Array** U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers** <28>.
 - Write 5hex and 007Fhex to **Diagnostics** U530 <33> and U210 Status and Mode Register.
 - Set **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
 - Toggle **Reset Control** U130D-11 WCARST(H) by writing 0000hex to **Address Decode/Select** port 1002hex <35>.
 - Write 01hex and 04hex to the **Compression Factor Counter** Register U824 <35> and the **Mode Select Latch** U724 <35>.
3. Manipulate the **MMU Gate Array** <33> to send an underrange data value (8001hex) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Status Read Back** buffer U822 <35> and verify that **Flags Decode** U100B-6 UNDER(H) <36> is low.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <35>.
7. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.

Error Index E53A1 The underrange data value is being decoded into the UNDER flag in the transparent mode when it shouldn't be.

Subsys Comm	WCA Control	Non Special (E53BX)
Routine Name	Non Special	
Overview	This test verifies that Flags Decode <36> does not generate special case flags (Null, Over and Under) for non-special data by sending the Compressor non-special data values.	
Description	<ol style="list-style-type: none"> 1. Initialize the MMU Gate Array U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/ Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the Interrupt Controllers <28>. <ul style="list-style-type: none"> • Write <i>5hex</i> and <i>007Fhex</i> to Diagnostics U530 <33> and U210 Status and Mode Register. • Set Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low. 2. Reset WCA to non-vectored test mode with a compression factor of 1. <ul style="list-style-type: none"> • Toggle Reset Control U130D-11 WCARST(H) by writing <i>0000hex</i> to Address Decode/Select port <i>1002hex</i> <35>. • Write <i>01hex</i> and <i>05hex</i> to the Compression Factor Counter Register U824 <35> and the Mode Select Latch U724 <35>. 3. Manipulate the MMU Gate Array <33> to send a non-special data value (<i>0001hex</i>) from the waveform RAM to the WCA. 4. Perform a software delay. 5. Read Status Read Back buffer U822 <35> and verify that U100C-8 NULL(H), U100A-3 OVER(H), and U100B-6 UNDER(H) are all low. 6. Repeat steps 2-5 for non-special case data values <i>0002hex</i>, <i>0004hex</i>, <i>0008hex</i>, <i>0010hex</i>, <i>0020hex</i>, <i>0040hex</i>, <i>0080hex</i>, <i>0100hex</i>, <i>0200hex</i>, <i>0400hex</i>, <i>0800hex</i>, <i>1000hex</i>, <i>2000hex</i>, and <i>4000hex</i>. 7. Reset the WCA by toggling Reset Control U130D-11 WCARST(H) <35>. 8. Disable SAG interrupt at the Interrupt Controllers <28> by setting Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low. 	
Error Index E53B1	The WCA decoded one of the non-special case data values as a special case, i.e., an Under, Over or Null condition was flagged.	

Executive

Routine Name Max Special

Overview This test verifies that the X Comparator & Y Comparator <36> produce the proper criterion for replacing the maximum value in X Min Max Latch Decode <36> & Y Min Max Latch Decode <36>. The comparator outputs are observed through Status Read Back buffer U622 <35> while special case data values (i.e., Null, Over and Under) are sent from the MMU Gate Array <33> to the Compressor.

Description

1. Initialize the MMU Gate Array U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/ Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the Interrupt Controllers <28>.
 - Write *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register.
 - Set Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 2.
 - Toggle Reset Control U130D-11 WCARST(H) <35> by writing *0000hex* to Address Decode/Select port *1002hex* <35>.
 - Write *02hex* and *05hex* to the Compression Factor Counter Register U824 <35> and the Mode Select Latch U724 <35>.
3. Manipulate the MMU Gate Array <33> to send a Null data value (*8000hex*) from the waveform RAM to the WCA. This becomes the "current" data value for the comparator's subsequent comparisons since this is the first data value after reset.
4. Perform a software delay.
5. Manipulate the MMU Gate Array <33> to send an Under data value (*8001hex*) from the waveform RAM to the WCA. This becomes the "input" data value for the comparator's comparison.
6. Read Status Read Back U622 <35> and verify that it reads *55hex*.

7. Repeat steps 2-6 for the following "current" and "input" combinations.

Current (<i>hex</i>) Data Value	Input (<i>hex</i>) Data Value	Expected Comparator Outputs (<i>hex</i>)
8000	7FFF	55
8000	8200	55
8001	8000	77
8001	7FFF	5A
8001	8200	5B
7FFF	8000	FF
7FFF	8001	F5
7FFF	8200	F5
8200	8000	DD
8200	8001	E5
8200	7FFF	5A

8. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
9. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0** low.

Error Index E5411

The output of the comparators did not match the known value. The display shows the first output that failed.

Routine Name Max DataLine (Max Data Lines)

Overview This test verifies the X Comparator <36> data lines by performing a "walking zero's" pattern test on the X Comparator <36> data lines.

Description

1. Initialize the MMU Gate Array U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/ Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the Interrupt Controllers <28>.
 - Write 5_{hex} and 007F_{hex} to Diagnostics U530 <33> and U210 Status and Mode Register.
 - Set Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 16.
 - Toggle Reset Control U130D-11 WCARST(H) <35> by writing 0000_{hex} to Address Decode/Select port 1002_{hex} <35>.
 - Write 10_{hex} and 05_{hex} to the Compression Factor Counter Register U824 <35> and the Mode Select Latch U724 <35>.
3. Manipulate the MMU Gate Array <33> to send the pattern 7FFE_{hex} from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read the Status Read Back U622 <35> and verify that it reads 55_{hex}.
6. Repeat steps 2-6 for the following patterns.

Input (hex) Data Value	Expected Comparator Outputs (hex)
7FFD	75
7FFB	75
7FF7	75
7FEF	75
7FDF	75
7FBF	75
7F7F	75
7EFF	E5
7DFF	E5
7BFF	E5
77FF	E5
6FFF	E5
5FFF	E5
3FFF	E5
FFFF	E5

8. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <35>.
9. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 0 low.

Error Index E5421

The output of the comparators did not match the known value. The display shows the first output that failed.

Routine Name Min DataLine (Min Data Lines)

Overview This test verifies Y Comparator <36> data lines by performing a "walking one's" pattern test on the Y Comparator <36> data lines.

Description

1. Initialize the MMU Gate Array U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the Interrupt Controllers <28>.
 - Write *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register.
 - Set Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 14.
 - Toggle Reset Control U130D-11 WCARST(H) <35> by writing *0000hex* to Address Decode/Select port *1002hex* <35>.
 - Write *0Ehex* and *05hex* to the Compression Factor Counter Register U824 <35> and the Mode Select Latch U724 <35>.
3. Manipulate the MMU Gate Array <33> to send the pattern *C000hex* from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read Status Read Back U622 <35> and verify that it reads *55hex*.
6. Repeat steps 2-6 for the following patterns.

Input (<i>hex</i>) Data Value	Expected Comparator Outputs (<i>hex</i>)
A000	5D
9000	5D
8800	5D
8400	5D
8200	5D
8100	5D
8080	5E
8040	5E
8020	5E
8010	5E
8008	5E
8004	5E
8002	5E

8. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
9. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28> Mask Register bit 1** and **Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0** low.

Error Index E5431

The output of the comparators did not match the known value. The display shows the first output that failed.

Routine Name Min Special

Overview This test verifies that the X Comparator & Y Comparator <36> produce the proper criterion for replacing the minimum value in X Min Max Latch Decode <36> & Y Min Max Latch Decode <36>. The comparator outputs are observed through Status Read Back buffer U622 <35> while special case data values (i.e. Null, Over and Under) are sent from the MMU Gate Array <33> to the Compressor.

Description

1. Initialize the MMU Gate Array U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the Interrupt Controllers <28>.
 - Write 5_{hex} and 007F_{hex} to Diagnostics U530 <33> and U210 Status and Mode Register.
 - Set Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 2 and switch the minimum and maximum registers.
 - Toggle Reset Control U130D-11 WCARST(H) <35> by writing 0000_{hex} to Address Decode/Select port 1002_{hex} <35>.
 - Write 02_{hex} and 05_{hex} to the Compression Factor Counter Register U824 <35> and the Mode Select Latch U724 <35>.
 - Set U332A-4 <35> low by writing 0000_{hex} to Address Decode/Select port 1008_{hex} <35>.
3. Manipulate the MMU Gate Array <33> to send a Null data value (8000_{hex}) from the waveform RAM to the WCA. This becomes the "current" data for the comparator's subsequent comparisons since this is the first data value after reset.
4. Perform a software delay.
5. Manipulate the MMU Gate Array <33> to send an Under data value (8001_{hex}) from the waveform RAM to the WCA. This becomes the "input" data value for the comparator's comparison.
6. Read the Status Read Back U622 <35> and verify that it reads 55_{hex}.

7. Repeat steps 2-6 for the following "current" and "input" combinations.

Current (<i>hex</i>) Data Value	Input (<i>hex</i>) Data Value	Expected Comparator Outputs (<i>hex</i>)
8000	7FFF	55
8000	8200	55
8001	8000	77
8001	7FFF	A5
8001	8200	B5
7FFF	8000	FF
7FFF	8001	5F
7FFF	8200	5F
8200	8000	DD
8200	8001	5E
8200	7FFF	A5

8. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H)** <35>.
9. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers MASTER U350** <28> Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360** <28> Mask Register bit 0 low.

Error Index E5441

The output of the comparators did not match the known value. The display shows the first comparator output that failed.

Routine Name Offset

Overview

This test verifies the **Adder Offset Register U624 & U722 <35>** by performing a "walking one's" test on this register. The "walking one" pattern in the Offset Register is added to a `0000hex` waveform data value and read back from **Adder Read Back <35>**.

Description

1. Initialize the **MMU Gate Array U210 <33>** and set it up to send test data to the Waveform Compressor/ Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers <28>**.
 - Write `5hex` and `007Fhex` to **Diagnostics U530 <33>** and **U210 Status and ModeRegister**.
 - Set **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1.
 - Toggle **Reset Control U130D-11 WCARST(H) <35>** by writing `0000hex` to **Adder Decode/Select port 1002hex**.
 - Write `01hex` and `04hex` to the **Compression Factor Counter Register U824 <35>** and the **Mode Select Latch U724 <35>**.
3. Manipulate the **MMU Gate Array <33>** to send a `0000hex` waveform value from the waveform RAM to the WCA.
4. Perform a software delay.
5. Write `0001hex` to **Adder Offset Register U624 & U722 <35>** of the WCA.
6. Read and verify the output (`0001hex`) of the **Adder** from **Adder Read Back <35>**.
7. Repeat steps 2-6 for patterns `0002hex`, `0004hex`, `0008hex`, `0010hex`, `0020hex`, `0040hex`, `0080hex`, `0100hex`, `0200hex`, `0400hex`, `0800hex`, `1000hex`, `2000hex`, `4000hex`, and `8000hex`.
8. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
9. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.

Error Index E5511

The pattern read from **Adder Read Back <35>** port did not match the pattern written to the **Adder Offset Register U624 & U722 <35>**. The display shows the first pattern that failed.

Executive

Routine Name Data Paths

Overview

This test verifies the data paths through MMU Input Register <36>, X Comp Output Latch <36>, Y Comp Output Latch <36>, Adder Input X Latch <37>, Adder Input Y Latch <37>, Adder Input Mux <37>, Adder <37>, Adder Output Select <37> and Adder Read Back <35> by performing a "walking one's" test on the data lines. In addition, the capability of Output Word Control <37> to recognize special data values (i.e., Null, Under and Over) is verified.

Description

1. Initialize the MMU Gate Array U210 <33> and set it up to send test data from the waveform RAM to the Waveform Compressor/Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the Interrupt Controllers <28>.
 - Write *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register.
 - Set Interrupt Controllers MASTER U350 <28> Mask Register bit 1 and Interrupt Controllers SLAVE 1 U360 <28> Mask Register bit 0 low.
2. Reset WCA to non-vectored test mode with a compression factor of 1 and an offset of 0.
 - Toggle Reset Control U130D-11 WCARST(H) <35> by writing *0000hex* to Adder Decode/Select port *1002hex*.
 - Write *01hex*, *05hex* and *0000hex* to the Compression Factor Counter Register U824 <35>, Mode Select Latch U724 <35>, and Adder Offset Register U624 & U722 <35>, respectively.
3. Manipulate the MMU Gate Array <33> to send pattern *0001hex* from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read and verify the data path through MMU Input Register <36>, X Comp Output Latch <36>, Adder Input X Latch <37>, Adder Input Mux <37>, Adder <37>, and Adder Output Select <37> by reading Adder Read Back <35> (value should be *0001hex*). Read Adder Read Back <35> once more to verify the path through Y Comp Output Latch <36>, Adder Input Y Latch <37> and Adder Input Mux <37> (the value should again be *0001hex*).
6. Repeat steps 2-5 with patterns *0002hex*, *0004hex*, *0008hex*, *0010hex*, *0020hex*, *0040hex*, *0080hex*, *0100hex*, *0200hex*, *0400hex*, *0800hex*, *1000hex*, *2000hex*, *4000hex*, *8000hex*, *8001hex*, and *7FFFhex*.
7. Reset the WCA by toggling Reset Control U130D-11 WCARST(H) <35>.

Executive

8. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers MASTER** U350 <28> Mask Register bit 1 and **Interrupt Controllers SLAVE 1** U360 <28> Mask Register bit 0 low.

Error Index E5521 The data value read from **Adder Read Back** <35> through the "X path" was not the same as the pattern sent from the MMU Gate Array <33>. The display shows the first pattern that failed.

Error Index E5522 The data value read from **Adder Read Back** <35> through the "Y path" was not the same as the pattern sent from the MMU Gate Array <33>. The display shows the first pattern that failed.

Routine Name Overrange

Overview This test verifies that the **Output Word Control <37>** causes **Adder Output Select <37>** to generate an overrange data value when the sum of the waveform data point sent from the **MMU Gate Array <33>** and the offset from the **Adder Offset Register <35>** exceeds an overrange value (7FFF_{hex}).

Description

1. Initialize the **MMU Gate Array U210 <33>** and set it up to send test data from the waveform RAM to the Waveform Compressor/ Adder (WCA). Enable Sequential Address Generator (SAG) interrupt at the **Interrupt Controllers <28>**.
 - Write 5_{hex} and 007F_{hex} to **Diagnostics U530 <33>** and U210 Status and Mode Register.
 - Set **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1 and an offset of 2.
 - Toggle **Reset Control U130D-11 WCARST(H) <35>** by writing 0000_{hex} to **Adder Decode/Select port 1002_{hex}**.
 - Write 01_{hex}, 04_{hex}, and 0002_{hex} to the **Compression Factor Counter Register U824 <35>**, **Mode Select Latch U724 <35>**, **Adder Offset Register U624 & U722 <35>**, respectively.
3. Manipulate the **MMU Gate Array <33>** to send an overrange data value (7FFF_{hex}) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Adder Read Back <35>** and verify that it is 7FFF_{hex}.
6. Reset the WCA by toggling **Reset Control U130D-11 WCARST(H) <35>**.
7. Disable SAG interrupt at the **Interrupt Controllers <28>** by setting **Interrupt Controllers MASTER U350 <28>** Mask Register bit 1 and **Interrupt Controllers SLAVE 1 U360 <28>** Mask Register bit 0 low.

Error Index E5531 The **Adder Output Select <37>** did not generate an overrange data value when the sum of waveform data point and the offset exceeded the overrange data value.

Routine Name Underrange

Overview This test verifies that the **Output Word Control** <37> causes **Adder Output Select** <37> to generate an underrange data value (8001hex) when the sum of the waveform data point sent from the **MMU Gate Array** <33> and the offset from **Adder Offset Register** <35> is below an underrange value.

Description

1. Initialize the **MMU Gate Array** U210 <33> and set it up to send test data from the waveform RAM to the **Waveform Compressor/ Adder (WCA)**. Enable **Sequential Address Generator (SAG)** interrupt at the **Interrupt Controllers** <28>.
 - Write 5hex and 007Fhex to **Diagnostics** U530 <33> and U210 Status and Mode Register.
 - Set **Interrupt Controllers MASTER** U350 <28> Mask Register bit 1 and **Interrupt Controllers SLAVE 1** U360 <28> Mask Register bit 0 low.
2. Reset WCA to transparent non-vectored test mode with a compression factor of 1 and an offset of – 4hex (FFFC_{hex}).
 - Toggle **Reset Control** U130D-11 WCARST(H) <35> by writing 0000hex to **Address Decode/Select** port 1002hex <35>.
 - Write 01hex, 04hex and FFFC_{hex} to the **Compression Factor Counter Register** U824 <35>, **Mode Select Latch** U724 <35> and **Adder Offset Register** U624 & U722 <35>, respectively.
3. Manipulate the **MMU Gate Array** <33> to send a waveform data word (8002hex) from the waveform RAM to the WCA.
4. Perform a software delay.
5. Read **Adder Read Back** <35> and verify that it is 8001hex.
6. Reset the WCA by toggling **Reset Control** U130D-11 WCARST(H) <35>.
7. Disable SAG interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers MASTER** U350 <28> Mask Register bit 1 and **Interrupt Controllers SLAVE 1** U360 <28> Mask Register bit 0 low.

Error Index E5541 The **Adder Output Select** <37> did not generate an underrange data value when the sum of the waveform data point and the offset is less than the underrange value.

Routine Name Display

Overview This test verifies the communication path from the Executive processor to the Display processor by performing a RAM test on the communication message area of waveform RAM and a "walking one's" and "walking zero's" pattern test on the data path through the Even & Odd DRAM <34>, Display Data Buffers <34>, and the Waveform Compressor/Adder (WCA) <36> <37>.

On power-up, this test is used by the Executive processor, after it has successfully completed its kernel testing, to detect the "presence" of the Display subsystem. In other words, whether the Display processor has successfully reached the communication stage of its kernel testing and can functionally communicate with the Executive processor through the previously mentioned hardware. At this time, when this test executes, the Display subsystem status is "not present" and the corresponding "Display Subsystem Not Present" section in the following description is the test path executed by the Executive processor.

Description

1. Initialize MMU Gate Array U210 <33> by writing *5hex* and *007Fhex* to Diagnostics U530 <33> and U210 Status and Mode Register.
2. Reset the WCA.
 - Toggle Reset Control U130D-11 WCARST(H) <35> by writing *0000hex* to Address Decode/Select port *1002hex* <35>.
 - Write *00hex*, *01hex*, and *0000hex* to the Mode Select Latch U724 <35>, Compression Factor Counter Register U824 <35>, and Adder Offset Register U722 & U624 <35>, respectively.
 - Toggle Reset Control U130D-11 WCARST(H) <35> by writing *0000hex* to Address Decode/Select port *1002hex* <35>.
3. Verify the waveform RAM memory locations (*7F000-7FFFFhex*) used for transmitting and receiving messages. Terminate test if any verify operation failed.
 - Fill the waveform RAM address range *7F000-7FFFFhex* with pattern *AAAAhex*.
 - Read and verify *7F000hex* for *AAAAhex*. If so, write *CCCChex* to *7F000hex*. Increment the address and continue this read/verify/write sequence until *7FFFEhex* is reached.
 - Repeat the read/verify/write sequence, starting again at *7FFFEhex* for *CCCChex*, *5555hex* and *0000hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
4. If there is no error in the waveform RAM, check to see if the Display subsystem is "present". If the Display subsystem was detected to be

"present" (i.e., have functional communication) on power-up by the Executive processor, then skip to step 10.

"Display Subsystem Not Present"

5. Fill 32 locations in the Display message area in the waveform RAM with "walking one's" and "walking zero's" patterns (see Error Index section below).
6. Program the MMU Gate Array U210 <33> Sequential Address Generator (SAG) to write a pattern to the Display and start the "write display" channel.
 - Set bit 3 of the Status and Mode Register (SMR) to clear any pending SAG interrupt.
 - Calculate the appropriate values for Message Length Register (MLR), Address Counter (AC), and Message Pointer Register (MPR) so that the SAG will transfer the waveform RAM location containing the pattern to be transmitted. Write the calculated values to the corresponding registers.
 - Start the "write display" channel by setting bit 9 of SMR.
 - Wait for the SAG to complete the transfer - repeatedly read SMR until bit 3 of SMR is high or a software timeout period expires.
 - Irrespective of the completion of the transfer, clear the SAG interrupt by setting bit 3 of SMR high.
7. Receive the pattern (bit-wise inversion of the transmitted pattern) from the Display subsystem.
 - Wait for the display talk request interrupt - repeatedly read SMR until bit 1 of SMR is high or a software timeout period expires.
 - If bit 1 of SMR is high, then clear the display talk request interrupt by setting bit 1 of SMR high.
 - Program the SAG to receive one word from the Display to the same location it was transmitted from.
 - Start the "read display" channel by setting bit 10 of SMR.
 - Wait for the SAG to complete the transfer - repeatedly read SMR until bit 3 of SMR is high or a software timeout period expires.
 - On completion of the transfer, clear the SAG interrupt by setting bit 3 of SMR high.
8. Read and verify the waveform RAM memory location from which the pattern was transmitted (see Error Index section below). If the received pattern was incorrect, then set the next pattern to be sent to DEAD_{hex}.

Executive

This will cause the Display subsystem, upon receiving the invalid pattern, to enter a continuous loop in which it inverts and echoes every pattern sent thereafter.

- 9. Repeat steps 6-8 for the remaining patterns and then terminate the test.

"Display Subsystem Present"

- 10. Enable MMU interrupts so that normal operating communications can take place.
- 11. Using the normal operating communications (similar to above but with multiple words and additional information), send a block of 32 words containing "walking one's" and "walking zero's" patterns (see Error Index section below) to the Display subsystem.
- 12. Using the normal operating communications, receive the block of bit-wise inverted patterns from the Display subsystem and verify that they are correct.
- 13. Disable MMU interrupts.

Error Index E5611

The pattern received from the Display is not the bit-wise inversion of the pattern sent. The display shows the first pattern that failed.

The patterns sent and those expected to be received are shown in the following table:

Pattern Sent (<i>hex</i>)	Expected Pattern Received (<i>hex</i>)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	EFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
EFFF	1000
F7FF	0800

Subsys Comm

MainFrm Comm

Display (E561X)

FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001

Error Index E5612	No pattern was received from the Display when a pattern was expected. The display shows the first pattern that was not received.
Error Index E5613	One or more of the patterns in the block of patterns received from the Display was incorrect. The display shows the first pattern that failed (see table above for patterns).
Error Index E5615	The waveform RAM used for transferring messages to and from the Display had a fault. The pattern read was not the pattern written. The display shows the first pattern that failed.
See Also	The Subsys Comm Waveform RAM Address/Data test (E523X) if the error index was E5615 in to order see the correspondence of the address and data bits to waveform RAM devices.

Executive

Routine Name Time Base

Overview This test verifies the communication path from the Executive processor to the Timebase processor by performing a RAM test on the communication message area of waveform RAM and a "walking one's" and "walking zero's" pattern test on the data path through the Even & Odd DRAM <34> and Digitizer Data Latches & Buffers <34>.

On power-up, this test is used by the Executive processor, after it has successfully completed its kernel testing, to detect the "presence" of the Timebase subsystem. In other words, whether the Timebase processor has successfully reached the communication stage of its kernel testing and can functionally communicate with the Executive processor through the previously mentioned hardware. At this time, when this test executes, the Timebase subsystem status is "not present" and the corresponding "Timebase Subsystem Not Present" section in the following description is the test path executed by the Executive processor.

Description

1. Initialize MMU Gate Array U210 <33> by writing 5hex and 007Fhex to Diagnostics U530 <33> and U210 Status and Mode Register.
2. Verify the waveform RAM memory locations (7F000-7FFFFhex) used for transmitting and receiving messages. Terminate test if any verify operation failed.
 - Fill the waveform RAM address range 7F000-7FFFFhex with pattern AAAAhex.
 - Read and verify 7F000hex for AAAAhex. If so, write CCCChex to 7F000hex. Increment the address and continue this read/verify/write sequence until 7FFFEhex is reached.
 - Repeat the read/verify/write sequence, starting again at 7FFFEhex for CCCChex, 5555hex and 0000hex (i.e., reading and verifying the previous pattern written and then writing the next pattern).
3. If there is no error in the waveform RAM, check to see if the Timebase subsystem is "present". If the Timebase subsystem was detected to be "present" (i.e., have functional communication) on power-up by the Executive processor, then skip to step 10.

"Timebase Subsystem Not Present"

4. Fill 32 locations in the Timebase transmit message area in the waveform RAM with "walking one's" and "walking zero's" patterns (see Error Index section below).
5. Set up MMU Gate Array <33> Random Address Generator (RAG) register 14 to receive messages in the Timebase receive message area in the waveform RAM.

6. Program the MMU Gate Array U210 <33> Sequential Address Generator (SAG) to write a pattern to the Timebase and start the "write digitizer" channel.
 - Set bit 3 of the Status and Mode Register (SMR) to clear any pending SAG interrupt.
 - Calculate the appropriate values for Message Length Register (MLR), Address Counter (AC), and Message Pointer Register (MPR) so that the SAG will transfer the waveform RAM location containing the pattern to be transmitted. Write the calculated values to the corresponding registers.
 - Start the "write digitizer" channel by setting bit 11 of SMR.
 - Wait for the SAG to complete the transfer - repeatedly read SMR until bit 3 of SMR is high or a software timeout period expires.
 - Irrespective of the completion of the transfer, clear the SAG interrupt by setting bit 3 of SMR high.
7. Receive the pattern (bit-wise inversion of transmitted pattern) from the Timebase subsystem.
 - Wait for the timebase end message interrupt - repeatedly read SMR until bit 4 of SMR is high or a software timeout period expires.
 - If bit 4 of SMR is high, then clear the timebase end message interrupt by setting bit 4 of SMR high.
8. Read and verify the waveform RAM memory location to which the Timebase subsystem should have written the inverted pattern. If the received pattern was incorrect, then set the next pattern to be sent to DEAD_{hex}. This will cause the Timebase subsystem, upon receiving this invalid pattern, to enter a continuous loop in which it inverts and echoes every pattern sent thereafter.
9. Repeat steps 6-8 for the remaining patterns and then terminate the test.

"Timebase Subsystem Present"

10. Enable MMU interrupts so that normal operating communications can take place.
11. Using the normal operating communications (similar to above but with multiple words and additional information), send a block of 32 words containing "walking one's" and "walking zero's" patterns (see Error Index section below) to the Timebase subsystem.
12. Using the normal operating communications, receive the block of bit-wise inverted patterns from the Timebase subsystem and verify that they are correct.

Executive

13. Disable MMU interrupts.

Error Index E5621

The pattern received from the Timebase is not the bit-wise inversion of the pattern sent. The display shows the first pattern that failed.

The patterns sent and those expected to be received are shown in the following table:

Pattern Sent (<i>hex</i>)	Expected Pattern Received (<i>hex</i>)
0001	FFFE
0002	FFFD
0004	FFFB
0008	FFF7
0010	FFEF
0020	FFDF
0040	FFBF
0080	FF7F
0100	FEFF
0200	FDFF
0400	FBFF
0800	F7FF
1000	FFFF
2000	DFFF
4000	BFFF
8000	7FFF
7FFF	8000
BFFF	4000
DFFF	2000
FFFF	1000
F7FF	0800
FBFF	0400
FDFF	0200
FEFF	0100
FF7F	0080
FFBF	0040
FFDF	0020
FFEF	0010
FFF7	0008
FFFB	0004
FFFD	0002
FFFE	0001

Error Index E5622

No pattern was received from the Timebase when a pattern was expected. The display shows the first pattern that was not received.

Error Index E5623

One or more of the patterns in the block of patterns received from the Timebase was incorrect. The display shows the first pattern that failed (see table above for pattern set).

Subsys Comm

MainFrm Comm

Time Base (E562X)

Error Index E5625

The waveform RAM used for transferring messages to and from the Timebase had a fault. The pattern read was not the pattern written. The display shows the first pattern that failed.

See Also

The Subsys Comm Waveform RAM Address/Data test (E523X) if the error index was E5625 in to order see the correspondence of the address and data bits to waveform RAM devices.

Routine Name TBC Intrpt (TBC Interrupt)

Overview This test verifies the ability of **Bank Decode/Select U430C-6 GPSIG <27>**, via J83-48 GPSIG <33>, to generate interrupts to the Timebase processor at **TBC-MMU Handshake Control (Input) U760A-5 INT3(H) <19>**. If the Timebase processor was "not present" at power-up, then an "option not present" error index is displayed.

Description

1. Check to see if the Timebase subsystem is "present". If it is "not present", then terminate test.
2. Enable MMU interrupts so that normal operating communications can take place.
3. Send a command to the Timebase processor to cause it to clear any pending interrupt at **TBC-MMU Handshake Control (Input) U760A-5 INT3(H) <19>** (i.e. set it to zero). The Timebase processor checks to see if this was successful.
4. Receive an acknowledgement message from the Timebase processor and verify that the Timebase processor indicated that it was able to clear the interrupt successfully.
5. Produce a Timebase processor interrupt (i.e. set **TBC-MMU Handshake Control (Input) U760A-5 INT3(H):H <19>**).
 - Toggle **Bank Decode/Select U430C-6 GPSIG <27>** by setting U520-15 high and then low.
6. Send a command to the Timebase processor to cause it to check if the interrupt at **TBC-MMU Handshake Control (Input) U760A-5 INT3(H) <19>** was set high.
7. Receive an acknowledgement message from the Timebase processor and verify that the Timebase processor indicated that the interrupt was indeed set high.
8. Disable MMU interrupts.

Error Index E5631 The Timebase processor, on command from the Executive processor, could not clear (i.e. set to zero) the interrupt at **TBC-MMU Handshake Control (Input) U760A-5 INT3(H) <19>**.

Error Index E5632 The Timebase processor did not recognize that the interrupt at **TBC-MMU Handshake Control (Input) U760A-5 INT3(H) <19>**, produced via the Executive processor, was set high.

Error Index ???? The Timebase processor was "not present" at power-up, therefore the test was not run.

Routine Name Interrupt

Overview This test verifies the **Serial Data Interface** SDI IC U330 <25> interrupt by checking that any pending interrupt can be cleared and an interrupt can be generated.

Description

1. Program the **Serial Data Interface** SDI IC U330 <25> to loopback mode.
 - Set bits 3 and 4 (Receiver Reset and Transmitter Reset bits) of the Miscellaneous Control Select 1 latch high.
 - Write 38hex and 00hex to the SDI Control Select Register and SDI Interrupt Output Mask Register.
 - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.
2. Verify that there is no SDI interrupt pending at the **Interrupt Controllers** <28>.
 - Read **Interrupt Controllers** SLAVE 1 U360 <25> INTERRUPT REQUEST REGISTER and verify that IR4 U360-22 is low.
3. If there is no pending interrupt, enable the SDI interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 4 low.
4. Enable the transmitter empty interrupt at the **Serial Data Interface** SDI IC U330 <25> by writing 01hex to the SDI Interrupt Output Mask Register.
5. Perform a software delay and then verify that the SDI interrupt did occur.
6. Disable the SDI interrupt at the **Interrupt Controllers** <28> by setting **Interrupt Controllers** MASTER U350 <28> Mask Register bit 1 and **Interrupt Controllers** SLAVE 1 U360 <28> Mask Register bit 4 high.
7. Program the **Serial Data Interface** SDI IC U330 <25> to normal mode.
 - Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch high.
 - Write 3Chex to the SDI Control Select Register.
 - Read SDI I/O Status Register to find out the state of Left Data Back (bit 5), Center Data Back (bit 3), and Right Data Back (bit 0) bits.

- Write a value to the SDI Control Select Register so that bit 2 is high and bits 3, 4, and 5 reflect the state of Left Data Back, Center Data Back, and Right Data Back, respectively.
- Write 0E_{hex} to SDI Interrupt Output Mask Register.
- Set Receiver Reset and Transmitter Reset bits of the Miscellaneous Control Select 1 latch low.

Error Index E5711 The SDI interrupt did not occur.

Error Index E5712 The SDI interrupt could not be cleared at the **Interrupt Controllers SLAVE 1** U360 <28>.

Dsy Control

ROM Location

U612 (D111X)

Routine Name U612

Overview This test verifies that ROM & Select U612 <39> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0008_{hex} and E000A_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.

Error Index D1111 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Error Index ???? The bytes at E0008_{hex} and E000A_{hex} were not complementary.

Display

Dsy Control

ROM Location

U602 (D112X)

Routine Name U602

Overview This test verifies that ROM & Select U602 <39> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0009_{hex} and E000B_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.

Error Index D1121 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Error Index ???? The bytes at E0009_{hex} and E000B_{hex} were not complementary.

Display

Routine Name U612

Overview This test verifies the integrity of ROM & Select U612 <39> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U612 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index D1211 The computed checksum did not match the stored checksum.

Error Index ???? The bytes at E0008*hex* and E000A*hex* were not complementary.

Routine Name U602

Overview This test verifies the integrity of ROM & Select U602 <39> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0009_{hex} and E000B_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U602 except the first two and verify it against the checksum stored in the first two bytes of the device.

Error Index D1221 The computed checksum did not match the stored checksum.

Error Index ???? The bytes at E0009_{hex} and E000B_{hex} were not complementary.

Routine Name	Data Lines
Overview	This test verifies the data lines from the CPU , through the μ P Data Buffers <39>, to the General Purpose Static RAM <39> by performing a "walking one's" test on one static RAM memory location.
Description	<div><div>1. Perform a "walking one's" test on General Purpose Static RAM <39> address 00000hex. Terminate test if any verify operation fails.</div><div><ul style="list-style-type: none">Write the pattern 0000hex to address 00000hex. Read the same address and verify that it was 0000hex. Continue this write/read/verify sequence with the patterns 0001hex, 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, 8000hex, and 0000hex.</div></div>
Error Index D1311	The pattern read from General Purpose Static RAM <39> address 00000hex was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to RAM devices is shown in the following table:

D0-D7	D8-D15
U611	U601

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of the **General Purpose Static RAM <39>** by performing a RAM test on all static RAM memory locations.

Description

1. Verify **VRS Min Plane DRAM** and **VRS Max Plane DRAM <42>** address range *7C000hex* to *7CFFFhex*. Terminate test if any verify operation fails.
 - Fill address range *7C000hex* to *7CFFFhex* with the pattern *AAAAhex*.
 - Read and verify address *7C000hex* for *AAAAhex*. If so, write *CCCChex* to address *7C000hex*. Increment the address and continue this read/verify/write sequence until address *7CFFFhex* is reached.
 - Repeat the read/verify/write sequence, starting again at address *7C000hex*, for *CCCChex*, *5555hex*, and *0000hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).
2. Save **General Purpose Static RAM <39>** data in address range *00000hex* to *00FFFhex* by copying it to VRS DRAM address range *7C000hex* to *7CFFFhex*.
3. Verify **General Purpose Static RAM <39>** address range *00000hex* to *00FFFhex*.
 - Perform the same procedures as in step 1 for the address range *00000hex* to *00FFFhex* using patterns *AAAAhex*, *CCCChex*, *F0F0hex*, *5555hex*, and *AAAAhex*.
 - Make one last read/verify pass for the pattern *AAAAhex* starting at address *00000hex*.
4. Copy the VRS DRAM data back from address range *7C000hex* to *7CFFFhex* to **General Purpose Static RAM <39>** address range *00000hex* to *00FFFhex*.

Error Index D1321

The pattern read from the displayed memory location in either the VRS DRAM or the **General Purpose Static RAM <39>** was not the pattern written. If the displayed address is between *7C000hex* and *7CFFFhex*, then the failure occurred in the VRS DRAM testing stage. Otherwise, the failure occurred while testing the static RAM. The correspondence of the address and data bits to RAM devices is shown in the following table:

Display

Dsy Control

Static RAM

Address/Data (D132X)

General Purpose Static RAM Address	D0-D7	D8-D15
00XXX _{hex}	U611	U601

X - don't care

See Also

See the Video Gen Trace Plane Address/Data (D252X) Error Index section for the correspondence of the address and data bits to VRS DRAM devices if the failure address displayed was between 7C000_{hex} and 7CFFF_{hex}.

Display

Routine Name Timer 0

Overview This test verifies CPU & Ready Logic U524 <39> Programmable Timers timer 0's counting accuracy and its interrupt. The counting accuracy of the timer 0 is verified by counting the system clock for a short duration. If the timer 0 counts accurately, a timer 0 interrupt is generated and verified.

Description

1. Enable the timer 0 to count the system clock by setting its gate signal high.
 - Set Diagnostic Control/Status Latch TMR IN 0 <43> high by writing 83hex to U600 <43>.
2. Stop the timer 0 by writing 4000hex to its Timer Mode/Control Register.
3. Program the counter to count up by writing 0000hex, FFFFhex, and C000hex to timer 0 Count Register, Max Count Value A Register, and Timer 0 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing 4000hex to its Timer Mode/Control Register.
6. Read and verify the timer 0 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 0 is counting accurately), enable the timer 0 to count the system clock again by setting its gate signal high. Else terminate testing.
 - Set Diagnostic Control/Status Latch TMR IN 0 <43> high by writing 83hex to U600 <43>.
8. Enable timer 0 interrupt by writing 0004hex to the Timer 0 Interrupt Control Register.
9. Program the counter to count up to 100hex and generate an interrupt when the count reaches 100hex.
 - Write 0100hex, 0000hex, E000hex to timer 0 Max Count Value A Register, Count Register, and Timer 0 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing 4000hex to its Timer Mode/Control Register.
12. Disable timer 0 interrupt by writing 000Chex to the Timer 0 Interrupt Control Register.
13. Verify that the timer 0 interrupt did occur.

Display

Dsy Control

Timers

Timer 0 (D141X)

- Error Index D1411 The timer 0 count was not within the expected tolerance.
- Error Index D1412 Timer 0 interrupt did not occur.

Display

Routine Name Timer 1

Overview

This test verifies CPU & Ready Logic U524 <39> Programmable Timers timer 1's counting accuracy and its interrupt. The counting accuracy of the timer 1 is verified by counting the system clock for a short duration. If the timer 1 counts accurately, a timer 1 interrupt is generated and verified.

Description

1. Enable the timer 1 to count the system clock by setting its gate signal high.
 - Set Diagnostic Control/Status Latch TMR IN 1 <43> high by writing *A3hex* to U600 <43>.
2. Stop the timer 1 by writing *4000hex* to its Timer Mode/Control Register.
3. Program the counter to count up by writing *0000hex*, *FFFFhex*, and *C000hex* to timer 1 Count Register, Max Count Value A Register, and Timer 1 Mode/Control Register, respectively.
4. Perform a software delay.
5. Stop the timer by writing *4000hex* to its Timer Mode/Control Register.
6. Read and verify the timer 1 count against the known value for the expected tolerance.
7. If the count was within the expected tolerance (i.e., the timer 1 is counting accurately), enable the timer 1 to count the system clock again by setting its gate signal high. Else terminate testing.
 - Set Diagnostic Control/Status Latch TMR IN 1 <43> high by writing *A3hex* to U600 <43>.
8. Enable timer 1 interrupt by writing *0004hex* to the Timer 1 Interrupt Control Register.
9. Program the counter to count up to *100hex* and generate an interrupt when the count reaches *100hex*.
 - Write *0100hex*, *0000hex*, *E000hex* to timer 1 Max Count Value A Register, Count Register, and Timer 1 Mode/Control Register, respectively.
10. Perform a software delay.
11. Stop the timer by writing *4000hex* to its Timer Mode/Control Register.
12. Disable timer 1 interrupt by writing *000Chex* to the Timer 1 Interrupt Control Register.
13. Verify that the timer 1 interrupt did occur.

Display

Dsy Control

Timers

Timer 1 (D142X)

Error Index D1421 The timer 1 count was not within the expected tolerance.

Error Index D1422 Timer 1 interrupt did not occur.

Display

Routine Name Timer 2

Overview This test verifies CPU & Ready Logic U524 <39> Programmable Timers timer 2's counting accuracy and its interrupt. The counting accuracy of the timer 2 is verified by counting the system clock for a short duration. If the timer 2 counts accurately, a timer 2 interrupt is generated and verified.

Description 1. Stop the timer 2 by writing 4000*hex* to its Timer Mode/Control Register.

2. Program the counter to count up by writing 0000*hex*, FFFF*hex*, and C000*hex* to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.

3. Perform a software delay.

4. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.

5. Read and verify the timer 2 count against the known value for the expected tolerance.

6. If the count was within the expected tolerance (i.e., the timer 2 is counting accurately), enable the timer 2 interrupt by writing 0004*hex* to the Timer 2 Interrupt Control Register. Else terminate testing.

7. Program the counter to count up to 100*hex* and generate an interrupt when the count reaches 100*hex*.

- Write 0100*hex*, 0000*hex*, E000*hex* to timer 2 Max Count Value A Register, Count Register, and Timer 2 Mode/Control Register, respectively.

8. Perform a software delay.

9. Stop the timer by writing 4000*hex* to its Timer Mode/Control Register.

10. Disable timer 2 interrupt by writing 000C*hex* to the Timer 2 Interrupt Control Register.

11. Verify that the timer 2 interrupt did occur.

Error Index D1431 The timer 2 count was not within the expected tolerance.

Error Index D1432 Timer 2 interrupt did not occur.

Routine Name DMA 0

Overview This test verifies CPU & Ready Logic U524 <39> Programmable DMA Unit's channel 0 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 0 interrupt by writing 0004_{hex} to DMA 0 Interrupt Control Register.
2. Initialize the source and destination memory locations in **General Purpose Static RAM** <39>.
 - Write patterns AAAA_{hex}, CCCC_{hex}, F0F0_{hex}, FF00_{hex} and 5555_{hex} to source memory locations starting at address 210_{hex}.
 - Write patterns 5555_{hex}, 3333_{hex}, 0F0F_{hex}, 00FF_{hex} and AAAA_{hex} to destination memory locations starting at address 200_{hex}.
3. Program the DMA channel 0 to transfer the 5 source patterns to the destination memory locations starting at 200_{hex} and generate an interrupt after completion of the transfer.
 - Write B725_{hex}, 0005_{hex}, 0000_{hex}, 0200_{hex}, 0000_{hex}, 0210_{hex} and B727_{hex} to DMA 0 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source Pointer Lower, and DMA 0 Channel Control Word registers, respectively. The last write initiates the DMA transfer.
4. Disable DMA 0 interrupt by writing 000C_{hex} to DMA 0 Interrupt Control Register.
5. Read and verify the destination memory locations.

Error Index D1511 One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Routine Name DMA 1

Overview This test verifies CPU & Ready Logic U524 <39> Programmable DMA Unit's channel 1 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Enable DMA 1 interrupt by writing 0004*hex* to DMA 1 Interrupt Control Register.
2. Initialize the source and destination memory locations in **General Purpose Static RAM <39>**.
 - Write patterns AAAA*hex*, CCC*hex*, F0F0*hex*, FF00*hex* and 5555*hex* to source memory locations starting at address 210*hex*.
 - Write patterns 5555*hex*, 3333*hex*, 0F0F*hex*, 00FF*hex* and AAAA*hex* to destination memory locations starting at address 200*hex*.
3. Program the DMA channel 1 to transfer the 5 source patterns to the destination memory locations starting at 200*hex* and generate an interrupt after completion of the transfer.
 - Write B725*hex*, 0005*hex*, 0000*hex*, 0200*hex*, 0000*hex*, 0210*hex* and B727*hex* to DMA 1 Channel Control Word, Transfer Count, Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, Source Pointer Lower, and DMA 1 Channel Control Word registers, respectively. The last write initiates the DMA transfer.
4. Disable DMA 1 interrupt by writing 000C*hex* to DMA 1 Interrupt Control Register.
5. Read and verify the destination memory locations.

Error Index D1521 One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Routine Name Command Port

Overview This test verifies the data path and operation of the Executive Processor Parallel Interface Port <43> and Interface Data Buffers <43> by utilizing Diagnostic Loopback Control <43> to loop "walking one" test patterns from the output of the Executive Processor Parallel Interface Port <43> back to its input. Even though the Diagnostic Loopback Control <43> is utilized, the communication handshake mechanics are very similar to those in normal operation (i.e., as if a message was actually being sent to and from the Executive processor).

Description

1. Enable diagnostic loopback capability.
 - Set Diagnostic Loopback Control LOOPBACKEN(L):L <43> (via MUART U523-27 P24 <39>).
2. Write the test pattern 0000hex to the Executive Processor Parallel Interface Port latches U731 & U733 <43>, through the Interface Data Buffers <43>.
3. Loop the test pattern from the Executive Processor Parallel Interface Port <43> outputs back to its inputs.
 - Toggle Diagnostic Loopback Control U623A-1 BMLCS(L). As BMLCS(L) goes low, the Executive Processor Parallel Interface Port outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches.
4. Read the test pattern from the Executive Processor Parallel Interface Port <43> through the Interface Data Buffers <43>.
5. Repeat steps 2-4 with the test patterns 0001hex, 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, and 8000hex.
6. Verify that each pattern read back was the same as the pattern written.
7. Disable diagnostic loopback capability.
 - Set Diagnostic Loopback Control LOOPBACKEN(L):H <43> (via MUART U523-27 P24 <39>).

Error Index D1611 The pattern read from the Executive Processor Parallel Interface Port <43> output port did not match the pattern written to its input port. The display shows the first pattern that did not match.

Routine Name DMA Access

Overview

This test verifies the operation of the CPU & Ready Logic 80186 Programmable DMA Unit <39> in reading and writing the Executive Processor Parallel Interface Port <43> by utilizing Diagnostic Loopback Control <43> to loop a test pattern from the output of the Executive Processor Parallel Interface Port <43> back to its input. Even though the Diagnostic Loopback Control <43> is utilized, the communication handshake mechanics are identical to those in normal operation (i.e., as if a message was actually being sent to and from the Executive processor).

Description

1. Enable diagnostic loopback capability.
 - Set Diagnostic Loopback Control LOOPBACKEN(L):L <43> (via MUART U523-27 P24 <39>).
2. Initialize the Executive Processor Parallel Interface Port <43> by reading port latches U731 & U733 through Interface Data Buffers <43>.
3. Write test pattern 5555_{hex} into General Purpose Static RAM <39> memory location 00210_{hex} (source location) and test pattern AAAA_{hex} into memory location 00204_{hex} (destination location).
4. Transfer pattern from General Purpose Static RAM <39> to the Executive Processor Parallel Interface Port <43>.
 - Program the CPU DMA 1 to read the value in General Purpose Static RAM <39> memory location 00210_{hex} and write it to the Executive Processor Parallel Interface Port <43>.
5. Transfer pattern from Executive Processor Parallel Interface Port <43> to the General Purpose Static RAM <39>.
 - Initialize CPU DMA 0 to read one value from the Executive Processor Parallel Interface Port <43> and write it to General Purpose Static RAM memory location 00204_{hex}.
 - Toggle Diagnostic Loopback Control U623A-1 BMLCS(L). As BMLCS(L) goes low, the Executive Processor Parallel Interface Port outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches. At this point, U733-5 DRQ0(H) becomes active (high) and initiates a DMA 0 transfer.
6. Read and verify that General Purpose Static RAM <39> memory location 00204_{hex} contains the pattern 5555_{hex}.
7. Disable diagnostic loopback capability.
 - Set Diagnostic Loopback Control LOOPBACKEN(L):H <43> (via MUART U523-27 P24 <39>).

Display

Dsy Control

Exec Intrfce

DMA Access (D162X)

Error Index D1621

The pattern read from **General Purpose Static RAM** <39> memory location 00204_{hex} (using a DMA 0 transfer from **Executive Processor Parallel Interface Port** <43>), was not the pattern written (5555_{hex}) to the **Executive Processor Parallel Interface Port** <43> (using a DMA 1 transfer from **General Purpose Static RAM** <39> memory location 00210_{hex}).

Display

Routine Name

Wavefrm Port (Waveform Port)

Overview

This test verifies the data path and operation of the **Executive Processor Parallel Interface Port <43>**, **Waveform Attribute Encoder <43>**, and **Waveform Data Buffers <43>** by utilizing **Diagnostic Loopback Control <43>** to loop test patterns from the output of the **Executive Processor Parallel Interface Port <43>** back to its input. Even though the **Diagnostic Loopback Control <43>** is utilized, the waveform handshake mechanics are very similar to those in normal operation (i.e., as if a waveform was actually being sent to the Display subsystem).

Description

1. Enable diagnostic loopback capability.
 - Set **Diagnostic Loopback Control** LOOPBACKEN(L):L <43> (via MUART U523-27 P24 <39>).
2. Initialize waveform attributes by writing *00hex* to MUART U523 (pins 32-37) P10-P17. This sets **Waveform Data Buffers** U622 CD9(H):L, CD13(H):L, CD14(H):L, and CD15(H):L <43>, and **Waveform Attribute Encoder** U533D-13 NULL_FORCE(H):L <43>, U635-1 SCALE(L):L, and U632-3 OFFSET(H):L.
3. Write a test pattern (see Error Index section below) to the **Executive Processor Parallel Interface Port** latches U731 & U733 <43>, through the **Interface Data Buffers <43>**.
4. Loop the test pattern from the **Executive Processor Parallel Interface Port <43>** outputs back to its inputs.
 - Toggle **Diagnostic Loopback Control** U623A-1 BMLCS(L). As BMLCS(L) goes low, the **Executive Processor Parallel Interface Port** outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches.
5. Read the test pattern from the **Executive Processor Parallel Interface Port <43>** through the **Waveform Attribute Encoder <43>** and **Waveform Data Buffers <43>**.
6. Repeat steps 3-5 for the remaining test patterns.
7. Verify that each pattern read back was the expected pattern (see Error Index section below).
8. Disable diagnostic loopback capability.
 - Set **Diagnostic Loopback Control** LOOPBACKEN(L):H <43> (via MUART U523-27 P24 <39>).

Error Index D1631

The pattern read from the **Executive Processor Parallel Interface Port <43>**, through the **Waveform Attribute Encoder <43>** and **Waveform Data Buffers <43>**, did not match the expected pattern. The display shows the first

Display

pattern that did not match. The patterns written to the **Executive Processor Parallel Interface Port <43>** and those patterns expected to be read back are shown in the following table:

Parallel Interface Port Pattern	Expected Waveform Data Buffer Pattern
AAAA _{hex}	0055 _{hex}
CCCC _{hex}	0099 _{hex}
F0F0 _{hex}	00E1 _{hex}
FF00 _{hex}	00FE _{hex}
5555 _{hex}	01AA _{hex}
7FFF _{hex} (Ovrrange)	09FF _{hex}
8000 _{hex} (Null)	1000 _{hex}
8001 _{hex} (Underrange)	0400 _{hex}

Routine Name

Attributes

Overview

This test verifies the waveform attribute control bits (color, age, and scale) from **MUART** (P10-P13) <39> through the **Waveform Data Buffers** <43> by performing a "walking one's" test on these lines. Waveform data test patterns are looped back through **Executive Processor Parallel Interface Port** <43>, **Waveform Attribute Encoder** <43>, and **Waveform Data Buffers** <43> by utilizing **Diagnostic Loopback Control** <43>.

Description

1. Enable diagnostic loopback capability.
 - Set **Diagnostic Loopback Control** LOOPBACKEN(L):L <43> (via **MUART** U523-27 P24 <39>).
2. Write a waveform data test pattern of 5555_{hex} to the **Executive Processor Parallel Interface Port** latches U731 & U733 <43>, through the **Interface Data Buffers** <43>.
3. Write a waveform attribute test pattern (see Error Index section below) to **MUART** U523 (pins 32-37) P10-P17 <39>.
4. Loop the waveform data test pattern from the **Executive Processor Parallel Interface Port** <43> outputs back to its inputs.
 - Toggle **Diagnostic Loopback Control** U623A-1 BMLCS(L). As BMLCS(L) goes low, the **Executive Processor Parallel Interface Port** outputs are enabled (U731-9 & U733-19). As BMLCS(L) goes high, the output data on DDB0-DDB15 is clocked back into the port input latches.
5. Read the waveform data and attribute test pattern from the **Executive Processor Parallel Interface Port** <43> through the **Waveform Attribute Encoder** <43> and **Waveform Data Buffers** <43>.
6. Repeat steps 2-5 for the remaining waveform attribute test patterns.
7. Verify that each pattern read back was the expected pattern (see Error Index section below).
8. Disable diagnostic loopback capability.
 - Set **Diagnostic Loopback Control** LOOPBACKEN(L):H <43> (via **MUART** U523-27 P24 <39>).

Error Index D1641

The pattern read from the **Executive Processor Parallel Interface Port** <43>, through the **Waveform Attribute Encoder** <43> and **Waveform Data Buffers** <43>, did not match the expected pattern. The display shows the first pattern that did not match. The waveform data patterns written to the **Executive Processor Parallel Interface Port** <43>, the waveform attribute patterns written to **MUART** P10-P17 <39>, and those patterns expected to be read back are shown in the following table:

Display

Dsy Control

Exec Intrfce

Attributes (D164X)

Parallel Interface Port
Waveform Data Pattern

MUART Waveform
Attribute Pattern

Expected Waveform
Data Buffer Pattern

5555*hex*

00*hex*

01AA*hex*

5555*hex*

01*hex*

21AA*hex*

5555*hex*

02*hex*

41AA*hex*

5555*hex*

04*hex*

81AA*hex*

5555*hex*

08*hex*

03AA*hex*

5555*hex*

40*hex*

00D5*hex*

Display

Routine Name Trace CAS

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **VRS Plane DRAM Control U223C-8 VRSCAS(L) <42>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-20 TMR IN 0 <39>**, where the number of transitions over a known period of time are counted.

VRSCAS(L) is a composite of other signals which are inherently verified as VRSCAS(L) is verified. The signals covered by VRSCAS(L) are **DRAM Control Generation U333C-8 CAS(H) <40>**, **Bit Plane Address Mux U325B-8 VA1(L) <40>**, **μP Address Latch U526-3 BHE(L) <39>**, **System Timing Generator U323D-8 ACCLK(H) <40>**, and **Video Memory Interface Synchronizer U430A-3 VGATE(L) <40>**.

Description

1. Enable **VRS Plane DRAM Control U223C-8 VRSCAS(L) <42>** to be counted.
 - Write *8Bhex* to **Diagnostic Control/Status Latch U600 <43>** to select VRSCAS(L) at the multiplexor inputs of **Diagnostic Timing Mux U433 <43>**. This also takes divide-by-four counter U516 out of reset.
2. Count the transitions of VRSCAS(L).
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 0 counter to start counting.
 - Perform a software delay.
 - Disable the timer 0 counter from counting.
3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2111

The value read from the **CPU & Ready Logic <39>** 80186 timer 0 counter was not within the expected tolerance limits (i.e. the measured frequency of VRSCAS(L) was out of tolerance).

Routine Name BSRLOAD

Overview This test verifies the relative frequency (with respect to the 80186 CPU) of **Video Shifter Control U425B-8 BSRLOAD(H) <40>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-20 TMR IN 0 <39>**, where the number of transitions over a known period of time are counted.

BSRLOAD(H) is a composite of other signals which are inherently verified as BSRLOAD(H) is verified. The signals covered by BSRLOAD(H) are **System Timing Generator U434B-4 PCLK(H)**, **U431-11 CCLK(H)**, **U431-12 FCLK(H)**, **U431-13 LCLK(H)**, **U431-14 MCLK(H) <40>** and **CRT Controller & Select U515-18 DISPLAY ENABLE(H) <40>**.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select U515 CRT CONTROLLER <40>**.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Enable **Video Shifter Control U425B-8 BSRLOAD(H) <40>** to be counted.
 - Write 9Bhex to **Diagnostic Control/Status Latch U600 <43>** to select BSRLOAD(H) at the multiplexor inputs of **Diagnostic Timing Mux U433 <43>**. This also takes divide-by-four counter U516 out of reset.
3. Perform a software delay of approximately 1 millisecond to wait for the CRT CONTROLLER U515 to finish the vertical retrace period (BSRLOAD(H) is only active while not in vertical retrace).
4. Count the transitions of BSRLOAD(H).
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 0 counter to start counting.
 - Perform a software delay.
 - Disable the timer 0 counter from counting.
5. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2121 The value read from the **CPU & Ready Logic <39>** 80186 timer 0 counter was not within the expected tolerance limits (i.e., the measured frequency of BSRLOAD(H) was out of tolerance).

Error Index D2122 A vertical sync condition in **CRT Controller & Select U515 <40>** could not be detected.

Routine Name Bit 1 CAS

Overview This test verifies the relative frequency (with respect to the 80186 CPU) of **Bit Plane 1 & 2 DRAM Control** U320B-6 BIT1CAS(L) <41> by routing it through **Diagnostic Timing Mux** <43> and into **CPU & Ready Logic** U524-20 TMR IN 0 <39>, where the number of transitions over a known period of time are counted.

BIT1CAS(L) is a composite of other signals which are inherently verified as BIT1CAS(L) is verified. The signals covered by BIT1CAS(L) are **DRAM Control Generation** U333C-8 CAS(H) <40>, **Bit Plane Address Mux** U325B-8 VA1(L) <40>, **μP Address Latch** U526-3 BHE(L) <39>, **System Timing Generator** U323D-8 ACCLK(H) <40>, and **Video Memory Interface Synchronizer** U430B-6 B1GATE(L) <40>.

Description

1. Enable **Bit Plane 1 & 2 DRAM Control** U320B-6 BIT1CAS(L) <41> to be counted.
 - Write AB_{hex} to **Diagnostic Control/Status Latch** U600 <43> to select BIT1CAS(L) at the multiplexor inputs of **Diagnostic Timing Mux** U433 <43>. This also takes divide-by-four counter U516 out of reset.
2. Count the transitions of BIT1CAS(L).
 - Write to an internal register of the **CPU & Ready Logic** 80186 to enable the internal timer 0 counter to start counting.
 - Perform a software delay.
 - Disable the timer 0 counter from counting.
3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2131 The value read from the **CPU & Ready Logic** <39> 80186 timer 0 counter was not within the expected tolerance limits (i.e., the measured frequency of BIT1CAS(L) was out of tolerance).

Routine Name ACCLK

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **System Timing Generator U323D-8 ACCLK(H) <40>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-20 TMR IN 0 <39>**, where the number of transitions over a known period of time are counted.

ACCLK(H) is a composite of other signals which are inherently verified as ACCLK(H) is verified. The signals covered by ACCLK(H) are **System Timing Generator U431-11 CCLK(H)** and **U431-14 MCLK(H)**.

Description

1. Enable **System Timing Generator U323D-8 ACCLK(H) <40>** to be counted.
 - Write BB_{hex} to **Diagnostic Control/Status Latch U600 <43>** to select ACCLK(H) at the multiplexor inputs of **Diagnostic Timing Mux U433 <43>**. This also takes divide-by-four counter U516 out of reset.
2. Count the transitions of ACCLK(H).
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 0 counter to start counting.
 - Perform a software delay.
 - Disable the timer 0 counter from counting.
3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2141

The value read from the **CPU & Ready Logic <39>** 80186 timer 0 counter was not within the expected tolerance limits (i.e., the measured frequency of ACCLK(H) was out of tolerance).

Routine Name Trace RAS

Overview

This test verifies the relative frequency (with respect to the 80186 CPU) of **System Timing Generator U222B-6 VRSRAS(L) <42>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-20 TMR IN 0 <39>**, where the number of transitions over a known period of time are counted.

VRSRAS(L) is a composite of other signals which are inherently verified as VRSRAS(L) is verified. The signals covered by VRSRAS(L) are **System Timing Generator U332C-8 RASE(H) <42>** and **DRAM Control Generation U333A-3 RAS(H) <42>**.

Description

1. Enable **VRS Plane DRAM Control U222B-6 VRSRAS(L) <42>** to be counted.
 - Write **CB_{hex}** to **Diagnostic Control/Status Latch U600 <43>** to select VRSRAS(L) at the multiplexor inputs of **Diagnostic Timing Mux U433 <43>**. This also takes divide-by-four counter U516 out of reset.
2. Count the transitions of VRSRAS(L).
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 0 counter to start counting.
 - Perform a software delay.
 - Disable the timer 0 counter from counting.
3. Read the timer 0 counter value and verify it is within expected limits.

Error Index D2151

The value read from the **CPU & Ready Logic <39> 80186 timer 0 counter** was not within the expected tolerance limits (i.e., the measured frequency of VRSRAS(L) was out of tolerance).

Routine Name CRTC R/W

Overview

This test verifies **CRT Controller & Select U631A-5 CRT R(H)/W(L) <40>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-21 TMR IN 1 <39>**, where the number of transitions are counted. CRT R(H)/W(L) goes high for every read of **CRT Controller & Select U515 <40>** and goes low for every write.

CRT R(H)/W(L) is a composite of other signals which are inherently verified as CRT R(H)/W(L) is verified. The signals covered by CRT R(H)/W(L) are **CPU & Ready Logic U524-27 CRTCS(L) <39>**, **U524-56 MPUCLK**, and **U533A-3 R(H)/W(L)**.

Description

1. Enable **CRT Controller & Select U631A-5 CRT R(H)/W(L) <40>** to be counted.
 - Write **93_{hex}** to **Diagnostic Control/Status Latch U600 <43>** to select CRT R(H)/W(L) at the multiplexor inputs of **Diagnostic Timing Mux U530 <43>**.
2. Clear **CRT Controller & Select U631A-5 CRT R(H)/W(L) <40>**.
 - Write to an internal register of **U515** to set CRT R(H)/W(L) low.
3. Enable **CPU & Ready Logic 80186 <39>** timer 1 counter.
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 1 counter to start counting from zero.
4. Generate and count occurrences of CRT R(H)/W(L).
 - Execute four read and write operations to internal registers of **CRT Controller & Select U515 <40>**. Each read operation sets **U631A-5 CRT R(H)/W(L)** high and the following write operation sets it back low, thereby causing the timer 1 counter to increment.
5. Disable the timer 1 counter from counting.
6. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2161

The value read from the **CPU & Ready Logic <39>** 80186 timer 1 counter did not equal the four read/write accesses made to **CRT Controller & Select U515 <40>** (i.e., CRT R(H)/W(L) did not toggle four times).

Routine Name B2HE

Overview This test verifies **Bit Plane 1 & 2 DRAM Control U323F-12 B2HE(L) <41>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-21 TMR IN 1 <39>**, where the number of transitions are counted. B2HE(L) toggles every time an access occurs to a high byte in **Bit Plane 2 DRAM <41>**.

B2HE(L) is a composite of other signals which are inherently verified as B2HE(L) is verified. The signals covered by B2HE(L) are **μP Address Latch U526-3 BHE(L) <39>** and **Video Memory Interface Synchronizer U430D-11 B2GATE(L) <40>**.

- Description**
1. Enable **Bit Plane 1 & 2 DRAM Control U323F-12 B2HE(L) <41>** to be counted.
 - Write **A3_{hex}** to **Diagnostic Control/Status Latch U600 <43>** to select B2HE(L) at the multiplexor inputs of **Diagnostic Timing Mux U530 <43>**.
 2. Enable **CPU & Ready Logic 80186 <39>** timer 1 counter.
 - Write to an internal register of the CPU & Ready Logic 80186 to enable the internal timer 1 counter to start counting from zero.
 3. Generate and count occurrences of B2HE(L).
 - Execute four read operations to **Bit Plane 2 DRAM <41>** location **60000_{hex}** (both low and high byte). Each read operation should toggle U323F-12 B2HE(L) and cause the timer 1 counter to increment.
 4. Disable the timer 1 counter from counting.
 5. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2171 The value read from the **CPU & Ready Logic <39>** 80186 timer 1 counter did not equal the four read accesses made to **Bit Plane 2 DRAM <41>** location **60000_{hex}** (i.e., B2HE(L) did not toggle four times).

Routine Name B1LE

Overview This test verifies **Bit Plane 1 & 2 DRAM Control U323B-4 B1LE(L) <41>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-21 TMR IN 1 <39>**, where the number of transitions are counted. B1LE(L) toggles every time an access occurs to a low byte in **Bit Plane 1 DRAM <41>**.

B1LE(L) is a composite of other signals which are inherently verified as B1LE(L) is verified. The signals covered by B1LE(L) are **μP Address Latch U614-19 MA0 <39>** and **Video Memory Interface Synchronizer U430B-6 B2GATE(L) <40>**.

- Description**
1. Enable **Bit Plane 1 & 2 DRAM Control U323B-4 B1LE(L) <41>** to be counted.
 - Write *B3hex* to **Diagnostic Control/Status Latch U600 <43>** to select B1LE(L) at the multiplexor inputs of **Diagnostic Timing Mux U530 <43>**.
 2. Enable **CPU & Ready Logic 80186 <39>** timer 1 counter.
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 1 counter to start counting from zero.
 3. Generate and count occurrences of B1LE(L).
 - Execute four read operations to **Bit Plane 1 DRAM <41>** location *50000hex* (both low and high byte). Each read operation should toggle U323B-4 B1LE(L) and cause the timer 1 counter to increment.
 4. Disable the timer 1 counter from counting.
 5. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2181 The value read from the **CPU & Ready Logic <39>** 80186 timer 1 counter did not equal the four read accesses made to **Bit Plane 1 DRAM <41>** location *50000hex* (i.e., B1LE(L) did not toggle four times).

Routine Name V0LE

Overview

This test verifies **VRS Plane DRAM Control U334D-8 V0LE(L) <42>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-21 TMR IN 1 <39>**, where the number of transitions are counted. V0LE(L) toggles every time an access occurs to a low byte in **VRS Min Plane DRAM <42>**.

V0LE(L) is a composite of other signals which are inherently verified as V0LE(L) is verified. The signals covered by V0LE(L) are **μP Address Latch U614-19 MA0 <39>**, **Video Memory Interface Synchronizer U430A-3 VGATE(L) <40>**, and **Bit Plane Address Mux U325B-9 VA1(H) <40>**.

Description

1. Enable **VRS Plane DRAM Control U334D-8 V0LE(L) <42>** to be counted.
 - Write **C3_{hex}** to **Diagnostic Control/Status Latch U600 <43>** to select V0LE(L) at the multiplexor inputs of **Diagnostic Timing Mux U530 <43>**.
2. Enable **CPU & Ready Logic 80186 <39>** timer 1 counter.
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 1 counter to start counting from zero.
3. Generate and count occurrences of V0LE(L).
 - Execute four read operations to **VRS Min Plane DRAM <42>** location **70000_{hex}** (both low and high byte). Each read operation should toggle U334D-8 V0LE(L) and cause the timer 1 counter to increment.
4. Disable the timer 1 counter from counting.
5. Read the timer 1 counter value and verify that it showed four occurrences.

Error Index D2191

The value read from the **CPU & Ready Logic <39>** 80186 timer 1 counter did not equal the four read accesses made to **VRS Min Plane DRAM <42>** location **70000_{hex}** (i.e., V0LE(L) did not toggle four times).

Routine Name Crastersync

Overview This test verifies the relative frequency (with respect to the 80186 CPU) of **CRT Controller & Select U515-39** CASTERSYNC(H) <40> by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-21** TMR IN 1 <39>, where the number of transitions over a known period of time are counted.

Description

1. Enable **CRT Controller & Select U515-39** CASTERSYNC(H) <40> to be counted.
 - Write **D3_{hex}** to **Diagnostic Control/Status Latch U600 <43>** to select CASTERSYNC(H) at the multiplexor inputs of **Diagnostic Timing Mux U530 <43>**.
2. Count the transitions of CASTERSYNC(H).
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 1 counter to start counting.
 - Perform a software delay.
 - Disable the timer 1 counter from counting.
3. Read the timer 1 counter value and verify it is within expected limits.

Error Index D21A1 The value read from the **CPU & Ready Logic <39>** 80186 timer 1 counter was not within the expected tolerance limits (i.e., the measured frequency of CASTERSYNC(H) was out of tolerance).

Routine Name Cfieldsync

Overview This test verifies the relative frequency (with respect to the 80186 CPU) of **CRT Controller & Select U515-40 CFIELDSYNC(H) <40>** by routing it through **Diagnostic Timing Mux <43>** and into **CPU & Ready Logic U524-21 TMR IN 1 <39>**, where the number of transitions over a known period of time are counted.

Description

1. Enable **CRT Controller & Select U515-40 CFIELDSYNC(H) <40>** to be counted.
 - Write **E3_{hex}** to **Diagnostic Control/Status Latch U600 <43>** to select **CFIELDSYNC(H)** at the multiplexor inputs of **Diagnostic Timing Mux U530 <43>**.
2. Count the transitions of **CFIELDSYNC(H)**.
 - Write to an internal register of the **CPU & Ready Logic 80186** to enable the internal timer 1 counter to start counting.
 - Perform a software delay.
 - Disable the timer 1 counter from counting.
3. Read the timer 1 counter value and verify it is within expected limits.

Error Index D21B1 The value read from the **CPU & Ready Logic <39> 80186 timer 1 counter** was not within the expected tolerance limits (i.e., the measured frequency of **CFIELDSYNC(H)** was out of tolerance).

Routine Name Dispen

Overview This test verifies the relative frequency (with respect to the 80186 CPU) of **CRT Controller & Select U515-18 DISPLAY ENABLE(H) <40>** by routing it through **Diagnostic Timing Mux <43>** and into CPU & Ready Logic U524-21 TMR IN 1 <39>, where the number of transitions over a known period of time are counted.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select U515 CRT CONTROLLER <40>**.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Enable **CRT Controller & Select U515-18 DISPLAY ENABLE(H) <40>** to be counted.
 - Write $F3_{hex}$ to **Diagnostic Control/Status Latch U600 <43>** to select DISPLAY ENABLE(H) at the multiplexor inputs of **Diagnostic Timing Mux U530 <43>**.
3. Perform a software delay of approximately 1 millisecond to wait for the CRT CONTROLLER U515 to finish the vertical retrace period (DISPLAY ENABLE(H) is only active while not in vertical retrace).
4. Count the transitions of DISPLAY ENABLE(H).
 - Write to an internal register of the CPU & Ready Logic 80186 to enable the internal timer 1 counter to start counting.
 - Perform a software delay.
 - Disable the timer 1 counter from counting.
5. Read the timer 1 counter value and verify it is within expected limits.

Error Index D21C1 The value read from the CPU & Ready Logic <39> 80186 timer 1 counter was not within the expected tolerance limits (i.e., the measured frequency of DISPLAY ENABLE(H) was out of tolerance).

Error Index D21C2 A vertical sync condition in **CRT Controller & Select U515 <40>** could not be detected.

Routine Name MPU Address

Overview This test verifies the address generation from the CPU and Ready Logic U524 CPU <39> through the Bit Plane Address Mux <40> and VRS Plane Address Mux <40> circuits by performing a "walking one's" test on the CPU generated address lines. The addresses from the CPU are used to access both the VRS planes and the bit planes. Addresses are captured and read back through the Diagnostic Bit & VRS Plane Address Feedback Latches <43>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the CRT Controller & Select U515 CRT CONTROLLER <40>.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Enable Diagnostic CRTC/MPU Address Trigger Control <43> to latch bit plane and VRS plane addresses.
 - Set Diagnostic Control/Status Latch U600-16 <43> high.
3. Reset Diagnostic CRTC/MPU Address Trigger Control <43>.
 - Read Diagnostic Bit & VRS Plane Address Feedback Latches U501 & U511 <43>. This drives U402A-4 low, which in turn sets U402A-5 high and allows U402B-9 to be driven high.
4. Read from an address in bit plane memory (see Error Index sections below).
 - Read from a bit plane memory location. The address generated by the CPU passes through the Bit Plane Address Mux <40> and VRS Plane Address Mux <40> and is intercepted by the Diagnostic Bit & VRS <43>. The address is latched in the normal row/column format. When an access is made to bit plane memory, Diagnostic CRTC/MPU Address Trigger Control U517A-2 GATE(H) <43> is driven high. This enables the row and column strobe signals at U401A-2 & U401D-12 <43>. U401A-3 Q6(H):L is the row strobe and U401D-11 CAS(L):L is the column strobe.
5. Read and verify latched bit plane address (see first Error Index section below).
 - Read Diagnostic Bit & VRS Plane Address Feedback Latches U500 & U510 <43>, ignoring U510-8 RA0 and U510-9 RA1 (their values are unpredictable), and verify against the expected address.
6. Read and verify latched VRS plane address (see second Error Index section below).

- Read Diagnostic Bit & VRS Plane Address Feedback Latches U501 & U511 <43>, ignoring U511-8 RA2 and U511-9 RA3 (their values are unpredictable), and verify against the expected address.
7. Repeat steps 3-6 until all patterns have been tested or a verify operation fails.

Error Index D2211

The bit plane latched address in Diagnostic Bit & VRS Plane Address Feedback Latches U500 & U510 did not match the expected value. The display shows the first bit plane address pattern which did not match. The CPU addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CPU Address	Expected Bit Plane Address
50000hex	0000hex
50001hex	0000hex
50002hex	0000hex
50004hex	0100hex
50008hex	0200hex
50010hex	0400hex
50020hex	0800hex
50040hex	1000hex
50080hex	2000hex
50100hex	4000hex
50200hex	8000hex
50400hex	0001hex
50800hex	0002hex
51000hex	0004hex
52000hex	0008hex
54000hex	0010hex
58000hex	0020hex

Video Gen

Address Mux

MPU Address (D221X)

Error Index D2212

The VRS plane latched address in **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 did not match the expected value. The display shows the first VRS plane address pattern which did not match. The CPU addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CPU Address	Expected VRS Plane Address
50000 _{hex}	0000 _{hex}
50001 _{hex}	0000 _{hex}
50002 _{hex}	0000 _{hex}
50004 _{hex}	1000 _{hex}
50008 _{hex}	2000 _{hex}
50010 _{hex}	4000 _{hex}
50020 _{hex}	8000 _{hex}
50040 _{hex}	0001 _{hex}
50080 _{hex}	0002 _{hex}
50100 _{hex}	0004 _{hex}
50200 _{hex}	0008 _{hex}
50400 _{hex}	0010 _{hex}
50800 _{hex}	0020 _{hex}
51000 _{hex}	0100 _{hex}
52000 _{hex}	0200 _{hex}
54000 _{hex}	0400 _{hex}
58000 _{hex}	0800 _{hex}

Error Index D2213

The vertical sync pulse from step 1 could not be detected within the timeout period.

Display

Routine Name

CRTC Address

Overview

This test verifies the address generation from the CRT Controller & Select U515 CRT CONTROLLER <40> through the Bit Plane Address Mapping <40>, Bit Plane Address Mux <40>, and VRS Plane Address Mux <40> circuits by performing a "walking one's" test on the CRT CONTROLLER generated address lines. Addresses are captured and read back through the Diagnostic Bit & VRS Plane Address Feedback Latches <43>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the CRT Controller & Select U515 CRT CONTROLLER <40>.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Enable Diagnostic CRTC/MPU Address Trigger Control <43> to latch bit plane and VRS plane addresses.
 - Set Diagnostic Control/Status Latch U600-16 <43> low.
3. Set CRT CONTROLLER output address (see Error Index sections below).
 - Wait for a vertical sync pulse as in step 1. After it is detected (i.e., during vertical retrace), set the address to be generated by writing the address into U515 CRT CONTROLLER's "top of page" register. This will be the first address generated after the CRT CONTROLLER finishes the vertical retrace period.
4. Reset Diagnostic CRTC/MPU Address Trigger Control <43>.
 - Read Diagnostic Bit & VRS Plane Address Feedback Latches U501 & U511 <43>. This drives U402A-4 low, which in turn sets U402A-5 high and allows U402B-9 to be driven high.
5. Wait for bit plane and VRS plane addresses to be latched into Diagnostic Bit & VRS Plane Address Feedback Latches <43>.
 - Wait for the next vertical sync pulse (as in step 1) to ensure that the bit plane and VRS plane addresses have been latched into U501, U511, U500, and U510. The bit plane address generated from the CRT CONTROLLER passes through Bit Plane Address Mapping <40>, Bit Plane Address Mux <40>, and into Diagnostic Bit & VRS Plane Address Feedback Latches U500 & U510 <43>. The VRS plane address passes only from the CRT CONTROLLER through the VRS Plane Address Mux and into latches U501 & U511 <43>. The addresses are latched in the normal row/column format. When the first address is generated by the CRT CONTROLLER, Diagnostic CRTC/MPU Address Trigger Control U402B-11 <43> is clocked. This enables the row and column strobe signals at U401A-2

& U401D-12 after U401B-6 CCLK(L) goes low. U401A-3 Q(H):L is the row strobe and U401D-11 CAS(L):L is the column strobe.

6. Read and verify latched bit plane address (see first Error Index section below).
 - Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U500 & U510 <43>, ignoring U510-8 RA0 and U510-9 RA1 (their values are unpredictable), and verify against the expected address.
7. Read and verify latched VRS plane address (see second Error Index section below).
 - Disable **Diagnostic CRTC/MPU Address Trigger Control** <43> from latching further bit plane and VRS plane addresses by setting **Diagnostic Control/Status Latch** U600-16 <43> high.
 - Read **Diagnostic Bit & VRS Plane Address Feedback Latches** U501 & U511 <43>, ignoring U511-8 RA2 and U511-9 RA3 (their values are unpredictable), and verify against the expected address.
8. Repeat steps 2-7 until all patterns have been tested or a verify operation fails.
9. Restore CRT CONTROLLER "top of page" register to its normal operating value (0000hex).

Error Index D2221

The bit plane latched address in **Diagnostic Bit & VRS Plane Address Feedback Latches** U500 & U510 did not match the expected value. The display shows the first bit plane address pattern which did not match. The CRT CONTROLLER addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CRT CONTROLLER Address	Expected Bit Plane Address
0000hex	0000hex
0001hex	0000hex
0002hex	0100hex
0004hex	0200hex
0008hex	0400hex
0010hex	0800hex
0020hex	1000hex
0040hex	0000hex
0080hex	0000hex
0100hex	8001hex
0200hex	0003hex
0400hex	0006hex
0800hex	000Chex
1000hex	0018hex
2000hex	0030hex
4000hex	0000hex
8000hex	0000hex

Display

Error Index D2222

The VRS plane latched address in Diagnostic Bit & VRS Plane Address Feedback Latches U501 & U511 did not match the expected value. The display shows the first VRS plane address pattern which did not match. The CRT CONTROLLER addresses generated and the expected addresses read back from U500 & U510 are shown in the following table:

CRT CONTROLLER Address	Expected VRS Plane Address
0000hex	0000hex
0001hex	0100hex
0002hex	0200hex
0004hex	0400hex
0008hex	0800hex
0010hex	0000hex
0020hex	0000hex
0040hex	0000hex
0080hex	0000hex
0100hex	0001hex
0200hex	0002hex
0400hex	0004hex
0800hex	0008hex
1000hex	0010hex
2000hex	0020hex
4000hex	0000hex
8000hex	0000hex

Error Index D2223

The vertical sync pulse from step 1 could not be detected within the timeout period.

Routine Name CRTC R/W

Overview This test verifies the capability to write and read the control registers of **CRT Controller & Select U515 CRT CONTROLLER <40>** by writing and reading one pattern to one CRT CONTROLLER control register.

Description

1. Read and save U515 CRT CONTROLLER <40> "cursor position low byte" control register.
2. Write test pattern to CRT CONTROLLER control register.
 - Write *55hex* to the CRT CONTROLLER "cursor position low byte" control register. This sets U534-1 & 2 CRTCS(L) and U631A-2 R/W(L) all low. When U631A-3 clocks, due to U534-8 MPUCLK clocking, U515-22 CRT R/W(L) goes low and the test pattern is written into the CRT CONTROLLER when U515-23 02(H) clocks.
3. Read and verify test pattern from CRT CONTROLLER control register.
 - Read from CRT CONTROLLER "cursor position low byte" control register. This sets U534-1 & 2 CRTCS(L) low and U631A-2 R/W(L) high. When U631A-3 clocks, due to U534-8 MPUCLK clocking, U515-22 CRT R/W(L) goes high and the control register is read from the CRT CONTROLLER when U515-23 02(H) clocks. The register value read is checked to see that it is *55hex*.
4. Restore the CRT CONTROLLER's "cursor position low byte" control register with the value saved in step 1.

Error Index D2231 The pattern read back from the **CRT Controller & Select U515 CRT CONTROLLER <40>** "cursor position low byte" control register was not the pattern that was written (i.e., *55hex*).

Routine Name

Data Lines

Overview

This test verifies the data lines from the CPU, through Video Memory Buffer & Select <39> and Plane 1 Data Buffer <41>, to the Bit Plane 1 DRAM <41> by performing a "walking one's" test on one bit plane 1 memory location.

Description

1. Perform a "walking one's" test on Bit Plane 1 DRAM <41> address 50000hex. Terminate test if any verify operation fails.

- Write the pattern 0000hex to address 50000hex. Read the same address and verify that it was 0000hex. Continue this write/read/verify sequence with the patterns 0001hex, 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, and 8000hex.

Error Index D2311

The pattern read from Bit Plane 1 DRAM <41> address 50000hex was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to DRAM devices is shown in the following table:

D0-D3	D4-D7	D8-D11	D12-D15
U213	U212	U211	U210

Routine Name Address/Data

Overview This test verifies the address lines and data integrity of the **Bit Plane 1** DRAM <41> by performing a RAM test on all bit plane 1 memory locations.

Description

1. Verify **Bit Plane 1** DRAM <41> address range 50000hex to 5FFFFhex. Terminate test if any verify operation fails.
 - Fill address range 50000hex to 5FFFFhex with the pattern AAAAhex.
 - Read and verify address 50000hex for AAAAhex. If so, write CCCChex to address 50000hex. Increment the address and continue this read/verify/write sequence until address 5FFFFhex is reached.
 - Repeat the read/verify/write sequence, starting again at address 50000hex, for CCCChex, 5555hex, and 0000hex (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index D2321 The pattern read from the displayed memory location in **Bit Plane 1** DRAM <41> was not the pattern written. The correspondence of the address and data bits to RAM devices is shown in the following table:

Bit Plane 1 DRAM Address	D0-D3	D4-D7	D8-D11	D12-D15
5XXX0hex, 5XXX4hex, 5XXX8hex, 5XXXChex	U213 U213	U212 U212	U211 U211	U210 U210
5XXX2hex, 5XXX6hex, 5XXXAhex, 5XXXEhex	U217 U217	U216 U216	U215 U215	U214 U214

X - don't care

Routine Name

Data Lines

Overview

This test verifies the data lines from the CPU, through Video Memory Buffer & Select <39> and Plane 2 Data Buffer <41>, to the Bit Plane 2 DRAM <41> by performing a "walking one's" test on one bit plane 2 memory location.

Description

1. Perform a "walking one's" test on Bit Plane 2 DRAM <41> address 60000hex. Terminate test if any verify operation fails.
 - Write the pattern 0000hex to address 60000hex. Read the same address and verify that it was 0000hex. Continue this write/read/verify sequence with the patterns 0001hex, 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex, and 8000hex.

Error Index D2411

The pattern read from Bit Plane 2 DRAM <41> address 60000hex was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to DRAM devices is shown in the following table:

D0-D3	D4-D7	D8-D11	D12-D15
U313	U312	U311	U310

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of the **Bit Plane 2 DRAM <41>** by performing a RAM test on all bit plane 2 memory locations.

Description

1. Verify **Bit Plane 2 DRAM <41>** address range *60000hex* to *6FFFFhex*. Terminate test if any verify operation fails.
 - Fill address range *60000hex* to *6FFFFhex* with the pattern *AAAAhex*.
 - Read and verify address *60000hex* for *AAAAhex*. If so, write *CCCChex* to address *60000hex*. Increment the address and continue this read/verify/write sequence until address *6FFFFhex* is reached.
 - Repeat the read/verify/write sequence, starting again at address *60000hex*, for *CCCChex*, *5555hex*, and *0000hex* (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index D2421

The pattern read from the displayed memory location in **Bit Plane 2 DRAM <41>** was not the pattern written. The correspondence of the address and data bits to RAM devices is shown in the following table:

Bit Plane 2 DRAM Address	D0-D3	D4-D7	D8-D11	D12-D15
<i>6XXX0hex, 6XXX4hex,</i>	U313	U312	U311	U310
<i>6XXX8hex, 6XXXChex</i>	U313	U312	U311	U310
<i>6XXX2hex, 6XXX6hex,</i>	U317	U316	U315	U314
<i>6XXXAhex, 6XXXEhex</i>	U317	U316	U315	U314

X - don't care

Routine Name

Data Lines

Overview

This test verifies the data lines from the CPU, through **Video Memory Buffer & Select <39>** and **VRS Plane Data Buffers <42>**, to the **VRS Min Plane DRAM <42>** by performing a "walking one's" test on one VRS plane memory location.

Description

1. Perform a "walking one's" test on **VRS Min Plane DRAM <42>** address `70000hex`. Terminate test if any verify operation fails.
 - Write the pattern `0000hex` to address `70000hex`. Read the same address and verify that it was `0000hex`. Continue this write/read/verify sequence with the patterns `0001hex`, `0002hex`, `0004hex`, `0008hex`, `0010hex`, `0020hex`, `0040hex`, `0080hex`, `0100hex`, `0200hex`, `0400hex`, `0800hex`, `1000hex`, `2000hex`, `4000hex`, and `8000hex`.

Error Index D2511

The pattern read from **VRS Min Plane DRAM <42>** address `70000hex` was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to DRAM devices is shown in the following table:

D0-D3	D4-D7	D8-D11	D12-D15
U110	U111	U112	U113

Routine Name

Address/Data

Overview

This test verifies the address lines and data integrity of the VRS Min Plane DRAM <42> and VRS Max Plane DRAM <42> by performing a RAM test on all VRS plane memory locations.

Description

- Verify VRS memory address range 70000hex to 7FFFFhex. Terminate test if any verify operation fails.
 - Fill address range 70000hex to 7FFFFhex with the pattern AAAAhex.
 - Read and verify address 70000hex for AAAAhex. If so, write CCCChex to address 70000hex. Increment the address and continue this read/verify/write sequence until address 7FFFFhex is reached.
 - Repeat the read/verify/write sequence, starting again at address 70000hex, for CCCChex, 5555hex, and 1000hex (i.e., reading and verifying the previous pattern written and then writing the next pattern).

Error Index D2521

The pattern read from the displayed memory location in VRS memory <42> was not the pattern written. VRS memory is divided into two banks, VRS Min Plane DRAM <42> and VRS Max Plane DRAM <42>. The correspondence of the address and data bits to RAM devices is shown in the following table:

VRS Memory Address	D0-D3	D4-D7	D8-D11	D12-D15
7XXX0hex, 7XXX4hex,	U110	U111	U112	U113
7XXX8hex, 7XXXChex	U110	U111	U112	U113
7XXX2hex, 7XXX6hex,	U120	U121	U122	U123
7XXXAhex, 7XXXEhex	U120	U121	U122	U123

X - don't care

Routine Name Single Axis

Overview This test verifies the single axis operation of **VRS Generation & Control** <42>. It functionally tests the operations and circuits necessary for displaying waveforms on a single axis; for correctly resolving color attribute overlays between waveforms; and for correctly resolving the underrange, overrange, and null conditions of waveform data. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane DRAM** <42>, representing waveform data for eight separate waveforms plus the clipping waveform, and then reading the individual color pixel information out of the **VRS Generation & Control** CUSTOM GATE ARRAY U125 <42>.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select** U515 CRT CONTROLLER <40>.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control** CUSTOM GATE ARRAY U125 <42> to single axis mode by setting **MUART** U523-33 SCALE(L):L <39>.
3. Initialize **VRS Min Plane DRAM** <42> and **VRS Max Plane DRAM** <42>.
 - Load **VRS Max Plane DRAM** and **VRS Min Plane DRAM** with test data (eight waveforms plus the clipping waveform). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **VRS Generation & Control** U124, U231, and U230 <42> operate in a normal manner).
4. Capture and read individual pixel color data.
 - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all bit plane data and use the data only from the VRS planes. This is accomplished by toggling **MUART** U523-28 VRS IN CLK <39> and serially clocking data into the input register through **MUART** U523-31 DATA OUT. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
 - Wait for two vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.
 - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking **MUART** U523-25 VRS OUT CLK and U523-26 VRS GATE <39>, reading 16 pixels, one-by-one, from U523-34 DATA IN.
5. Repeat steps 3 and 4 with a second set of test data.

Display

Video Gen

VRS Gen

Single Axis (D261X)

6. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2611

One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2612

A vertical sync condition in CRT Controller & Select U515 <40> could not be detected.

Display

Routine Name Dual Axis

Overview This test verifies the dual axis operation of **VRS Generation & Control <42>**. It functionally tests the operations and circuits necessary for displaying waveforms on a dual axis; for correctly resolving color attribute overlays between waveforms; and for correctly resolving the underrange, overrange, and null conditions of waveform data. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane DRAM <42>**, representing waveform data for eight separate waveforms plus the clipping waveform, and then reading the individual color pixel information out of the **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>**.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select U515 CRT CONTROLLER <40>**.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>** to dual axis mode by setting **MUART U523-33 SCALE(L):H <39>**.
3. Initialize **VRS Min Plane DRAM <42>** and **VRS Max Plane DRAM <42>**.
 - Load **VRS Max Plane DRAM** and **VRS Min Plane DRAM** with test data (eight waveforms plus the clipping waveform). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **VRS Generation & Control U124, U231, and U230 <42>** operate in a normal manner).
4. Capture and read individual pixel color data.
 - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all bit plane data and use the data only from the VRS planes. This is accomplished by toggling **MUART U523-28 VRS IN CLK <39>** and serially clocking data into the input register through **MUART U523-31 DATA OUT**. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
 - Wait for two vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.
 - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking **MUART U523-25 VRS OUT CLK** and **U523-26 VRS GATE <39>**, reading 16 pixels, one-by-one, from **U523-34 DATA IN**.
5. Repeat steps 3 and 4 with a second set of test data.

Display

Video Gen

VRS Gen

Dual Axis (D262X)

6. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2621

One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2622

A vertical sync condition in CRT Controller & Select U515 <40> could not be detected.

Display

Routine Name Color Map

Overview This test verifies the pixel color encoding operation of **VRS Generation & Control <42>**. It functionally tests the operations and circuits necessary for correctly resolving color attribute overlays between waveforms. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane DRAM <42>**, representing waveform data for eight separate waveforms plus the clipping waveform, and then reading the individual color pixel information out of the **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>**.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select U515 CRT CONTROLLER <40>**.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>** to single axis mode by setting **MUART U523-33 SCALE(L):L <39>**.
3. Initialize **VRS Min Plane DRAM <42>** and **VRS Max Plane DRAM <42>**.
 - Load **VRS Max Plane DRAM** and **VRS Min Plane DRAM** with test data (eight waveforms plus the clipping waveform). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **VRS Generation & Control U124, U231, and U230 <42>** operate in a normal manner).
4. Load the color map in the CUSTOM GATE ARRAY.
 - Toggle **MUART U523-29 COLOR CLK <39>** and serially clock data into the color map through **MUART U523-31 DATA OUT**.
5. Capture and read individual pixel color data.
 - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all bit plane data and use the data only from the VRS planes. This is accomplished by toggling **MUART U523-28 VRS IN CLK <39>** and serially clocking data into the input register through **MUART U523-31 DATA OUT**. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
 - Wait for 2 vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.

Video Gen

VRS Gen

Color Map (D263X)

- Read the serial pixel data from the CUSTOM GATE ARRAY by clocking MUART U523-25 VRS OUT CLK and U523-26 VRS GATE <39>, reading 16 pixels, one-by-one, from U523-34 DATA IN.

6. Repeat steps 4 and 5 with a second color map setting.
7. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2631

One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2632

A vertical sync condition in CRT Controller & Select U515 <40> could not be detected.

Display

Routine Name Video Shfter (Video Shifter)

Overview This test verifies **Plane 1 Video Shifter <41>** and **Plane 2 Video Shifter <41>**. **Bit Plane 1 DRAM <41>** and **Bit Plane 2 DRAM <41>** are loaded with data designed to display 16 consecutive pixels with an incremental color index. The test is designed to drive all bits high and low through the video shifter outputs. The test also functionally verifies the color attribute overlays between bit plane color indices. The color overlay function resides in the **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>**.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select U515 CRT CONTROLLER <40>**.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>** to single axis mode by setting **MUART U523-33 SCALE(L):L <39>**.
3. Initialize Bit Plane 1 DRAM <41> and Bit Plane 2 DRAM <41>.
 - Load **Bit Plane 1 DRAM** and **Bit Plane 2 DRAM** with test data (values *AAAAhex* and *5555hex*, respectively). The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **Plane 1 Video Shifter <41>**, **Plane 2 Video Shifter <41>**, and **VRS Generation & Control U124, U231, and U230 <42>** all operate in a normal manner).
4. Capture and read individual pixel color data.
 - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY to mask off all VRS plane data and use the data only from the bit planes. This is accomplished by toggling **MUART U523-28 VRS IN CLK <39>** and serially clocking data into the input register through **MUART U523-31 DATA OUT**. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
 - Wait for 2 vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.
 - Read the serial pixel data from the CUSTOM GATE ARRAY by clocking **MUART U523-25 VRS OUT CLK** and **U523-26 VRS GATE <39>**, reading 16 pixels, one-by-one, from **U523-34 DATA IN**.
5. Repeat steps 4 and 5 with a second set of test data (*5555hex* for **Bit Plane 1 DRAM** and *AAAAhex* for **Bit Plane 2 DRAM**).

Video Gen

VRS Gen

Video Shfter (D264X)

6. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2641

One of the three pixel color data values in one of the two test cases did not match the expected results.

Error Index D2642

A vertical sync condition in CRT Controller & Select U515 <40> could not be detected.

Display

Routine Name

Priority

Overview

This test verifies the priority encoding operation between bit plane and VRS plane color pixel data of **VRS Generation & Control <42>**. This is accomplished by placing data in the **VRS Max Plane DRAM** and **VRS Min Plane DRAM <42>**, representing waveform data for eight separate waveforms plus the clipping waveform, and by placing data in **Bit Plane 1 DRAM** and **Bit Plane 2 DRAM <41>** designed to display 16 consecutive pixels with incremental color indices. The priority encoder should override all waveform pixel data (from the VRS planes) with bit plane pixel data which contains a color index greater than zero. The individual color pixel information is read out of the **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>**.

Description

1. Wait for a vertical sync pulse (i.e., start of vertical retrace) from the **CRT Controller & Select U515 CRT CONTROLLER <40>**.
 - Repeatedly read the CRT CONTROLLER status register, waiting for either vertical sync to be asserted or a software timeout period to expire. If a vertical sync condition is not detected, generate an error and terminate test.
2. Initialize **VRS Generation & Control CUSTOM GATE ARRAY U125 <42>** to single axis mode by setting **MUART U523-33 SCALE(L):L <39>**.
3. Initialize **VRS Min Plane DRAM <42>**, **VRS Max Plane DRAM <42>**, **Bit Plane 1 DRAM <41>**, and **Bit Plane 2 DRAM <41>**.
 - Load **VRS Min Plane DRAM <42>**, **VRS Max Plane DRAM <42>**, **Bit Plane 1 DRAM <41>**, and **Bit Plane 2 DRAM <41>** with test data. The test data is then automatically loaded into the CUSTOM GATE ARRAY through the normal display mechanism of the CRT CONTROLLER (i.e., **Plane 1 Video Shifter <41>**, **Plane 2 Video Shifter <41>**, and **VRS Generation & Control U124, U231, and U230 <42>** all operate in a normal manner).
4. Capture and read individual pixel color data.
 - For each individual color (red, green, blue), set the diagnostic input register inside the CUSTOM GATE ARRAY so that it does not mask off either the VRS plane or bit plane data when capturing pixel information. This is accomplished by toggling **MUART U523-28 VRS IN CLK <39>** and serially clocking data into the input register through **MUART U523-31 DATA OUT**. The pixel color data is captured in a diagnostic serial shift register inside the CUSTOM GATE ARRAY.
 - Wait for two vertical sync pulses from the CRT CONTROLLER to guarantee that the pixel data has been captured.

Display

Video Gen

VRS Gen

Priority (D265X)

- Read the serial pixel data from the CUSTOM GATE ARRAY by clocking MUART U523-25 VRS OUT CLK and U523-26 VRS GATE <39>, reading 16 pixels, one-by-one, from U523-34 DATA IN.

5. Verify that all color pixel data captured from the CUSTOM GATE ARRAY matches expected values.

Error Index D2651

One of the three pixel color data values did not match the expected results.

Error Index D2652

A vertical sync condition in CRT Controller & Select U515 <40> could not be detected.

Display

Routine Name	Stimulus
Overview	This test verifies Z Axis Amplifier U601 and U600 <44> by changing the intensity of the display CRT screen via MUART U523 <39> through a set of walking one's patterns <i>01hex</i> , <i>02hex</i> , ..., <i>80hex</i> .
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector status set to "Off". Once this test is invoked, the operator can use an oscilloscope to view signals in the Z Axis Amplifier .
Description	<ol style="list-style-type: none">1. Read and save the value of MUART PORT 2.2. Set the Z Axis Amplifier U601 <44> (intensity or brightness register) to <i>01hex</i>.<ul style="list-style-type: none">• Read and save the value of MUART PORT 2 again.• Shift <i>01hex</i> into U601 bit by bit starting with the MSB (bit 7). In order to shift a bit into U601, the value of the bit is gated onto DATA OUT U523-31 <39> (by writing it to bit 0 of MUART PORT 2) and at the same time BRITE CLK U523-30 <39> is set low and then high (by toggling bit 1 of MUART PORT 2 while maintaining the value of bit 0).• Restore the value of MUART PORT 2.3. Repeat step 2 with values <i>02hex</i>, <i>04hex</i>, <i>08hex</i>, <i>10hex</i>, <i>20hex</i>, <i>40hex</i>, and <i>80hex</i>.4. Set the intensity register U601 back to the value it was before this test was executed.5. Restore the value of MUART PORT 2 again.
Error Index	None.

Routine Name U300

Overview This test verifies that ROM & Select U300 <17> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations C0008hex and C000Ahex and verify that the result is FFhex.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.

Error Index T1111 The bytes at C0008hex and C000Ahex were not complementary.

Error Index T1112 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

TBC Control

ROM Location

U310 (T112X)

Routine Name U310

Overview This test verifies that **ROM & Select** U310 <17> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations *C0009hex* and *C000Bhex* and verify that the result is *FFhex*.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.

Error Index T1121 The bytes at *C0009hex* and *C000Bhex* were not complementary.

Error Index T1122 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Routine Name	U400
--------------	------

Overview	This test verifies that ROM & Select U400 <17> is in the correct socket.
----------	--

Description	<ol style="list-style-type: none"> Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits. <ul style="list-style-type: none"> Exclusive-or the contents of byte locations E0008hex and E000Ahex and verify that the result is FFhex. If the complement test was successful, then verify the location byte of the EPROM device. <ul style="list-style-type: none"> Read and verify the location byte against the known value.
-------------	---

Error Index T1131	The bytes at E0008hex and E000Ahex were not complementary.
-------------------	--

Error Index T1132	The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.
-------------------	---

TBC Control

ROM Location

U410 (T114X)

Routine Name U410

Overview This test verifies that ROM & Select U410 <17> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0009hex and E000Bhex and verify that the result is FFhex.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.

Error Index T1141 The bytes at E0009hex and E000Bhex were not complementary.

Error Index T1142 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Timebase

Routine Name U300

Overview This test verifies the integrity of ROM & Select U300 <17> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations C0008*hex* and C000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U300 except the first two, and then verify it against the checksum stored in the first two bytes of the device.

Error Index T1211 The bytes at C0008*hex* and C000A*hex* were not complementary.

Error Index T1212 The computed checksum did not match the stored checksum.

Routine Name U310

Overview This test verifies the integrity of ROM & Select U310 <17> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations C0009_{hex} and C000B_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U310 except the first two, and then verify it against the checksum stored in the first two bytes of the device.

Error Index T1221 The bytes at C0009_{hex} and C000B_{hex} were not complementary.

Error Index T1222 The computed checksum did not match the stored checksum.

Routine Name U400

Overview This test verifies the integrity of ROM & Select U400 <17> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0008*hex* and E000A*hex* and verify that the result is FF*hex*.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U400 except the first two, and then verify it against the checksum stored in the first two bytes of the device.

Error Index T1231 The bytes at E0008*hex* and E000A*hex* were not complementary.

Error Index T1232 The computed checksum did not match the stored checksum.

Routine Name U410

Overview This test verifies the integrity of ROM & Select U410 <17> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations E0009hex and E000Bhex and verify that the result is FFhex.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U410 except the first two, and then verify it against the checksum stored in the first two bytes of the device.

Error Index T1241 The bytes at E0009hex and E000Bhex were not complementary.

Error Index T1242 The computed checksum did not match the stored checksum.

Routine Name

Data Lines

Overview

This test verifies **Ram Select** <17> and the data lines from CPU U440 <17>, through the **μP Kernel Data Buffers** <17>, to the **General Purpose Static RAM** <17> by performing a "walking one's" test on two static RAM memory locations. One location is for the low half of the RAM and the other location is for the high half of the RAM.

Description

1. Initialize the test locations.
 - Write the pattern *FFFFhex* to address *00000hex* and *10000hex*.
2. Verify **Ram Select** <17>.
 - Read address *00000hex* and verify that it is *FFFFhex*.
3. Perform a "walking one's" test on **General Purpose Static RAM** <17> address *00000hex*. Terminate test if any verify operation fails.
 - Write the pattern *8000hex* to address *00000hex*. Read the same address and verify that it was *8000hex*. Continue this write/read/verify sequence with the patterns *4000hex*, *2000hex*, *1000hex*, *0800hex*, *0400hex*, *0200hex*, *0100hex*, *0080hex*, *0040hex*, *0020hex*, *0010hex*, *0008hex*, *0004hex*, *0002hex*, *0001hex*.
4. Verify **Ram Select** <17>.
 - Write *0000hex* to address *00000hex*.
 - Read address *10000hex* and verify that it is *FFFFhex*.
5. Perform a "walking one's" test on **General Purpose Static RAM** <17> address *10000hex*. Terminate test if any verify operation fails.
 - Write the pattern *8000hex* to address *10000hex*. Read the same address and verify that it was *8000hex*. Continue this write/read/verify sequence with the patterns *4000hex*, *2000hex*, *1000hex*, *0800hex*, *0400hex*, *0200hex*, *0100hex*, *0080hex*, *0040hex*, *0020hex*, *0010hex*, *0008hex*, *0004hex*, *0002hex*, *0001hex*.
6. Verify **Ram Select** <17>.
 - Write *0000hex* to address *10000hex*.
 - Read address *00000hex* and address *10000hex* and verify that both are *0000hex*.

Error Index T1311

The pattern read from **General Purpose Static RAM** <17> address *00000hex* or address *10000hex* was not the same pattern written. The display shows the first pattern that failed. The correspondence of the data bits to RAM devices is shown in the following table:

Timebase

TBC Control

Static RAM

Data Lines (T131X)

Address	D0-D7	D8-D15
00000 _{hex}	U501	U510
10000 _{hex}	U500	U511

Timebase

Routine Name	Address/Data
Overview	This test verifies the address lines and data integrity of the General Purpose Static RAM <17> by performing a RAM test on all static RAM memory locations.
Description	<ol style="list-style-type: none"> 1. Verify General Purpose Static RAM <17> address range <code>10000hex</code> to <code>13FEFhex</code>. Terminate test if any verify operation fails. <ul style="list-style-type: none"> • Fill address range <code>10000hex</code> to <code>13FEFhex</code> with the pattern <code>AAAAhex</code>. • Read and verify address <code>10000hex</code> for <code>AAAAhex</code>. If so, write <code>CCCChex</code> to address <code>10000hex</code>. Increment the address and continue this read/verify/write sequence until address <code>13FEFhex</code> is reached. • Repeat the read/verify/write sequence, starting again at address <code>10000hex</code>, for <code>F0F0hex</code>, <code>5555hex</code>, and <code>AAAAhex</code> (i.e., reading and verifying the previous pattern written and then writing the next pattern). • Make one last read/verify pass for the pattern <code>AAAAhex</code> starting at address <code>10000hex</code>. 2. Save General Purpose Static RAM <17> data in address range <code>00000hex</code> to <code>03FEFhex</code> by copying it to address range <code>10000hex</code> to <code>13FEFhex</code>. 3. Verify General Purpose Static RAM <17> address range <code>00000hex</code> to <code>0FFFFhex</code>. <ul style="list-style-type: none"> • Perform the same procedures as in step 1. 5. Copy the General Purpose Static RAM <17> data back from address range <code>10000hex</code> to <code>1FFFFhex</code> to address range <code>00000hex</code> to <code>0FFFFhex</code>. 6. Verify General Purpose Static RAM <17> address range <code>10000hex</code> to <code>1FFFFhex</code>. <ul style="list-style-type: none"> • Perform the same procedures as in step 1. 7. Copy the General Purpose Static RAM <17> data back from address range <code>03FF0hex</code> to <code>0FFFFhex</code> to address range <code>13FF0hex</code> to <code>1FFFFhex</code>.
Error Index T1321	The pattern read from the displayed memory location in the General Purpose Static RAM <17> was not the pattern written. The correspondence of the address and data bits to RAM devices is shown in the following table:

TBC Control

Static RAM

Address/Data (T132X)

Address	D0-D7	D8-D15
0XXXX _{hex}	U501	U510
1XXXX _{hex}	U500	U511
X - don't care		

Timebase

Routine Name Battery

Overview This test verifies the battery back-up capability of the **General Purpose Static RAM** <17> sockets for U500 & U511 by checking that four confidence words are correct. This part of the static RAM is essentially non-volatile.

Description

1. Verify the four confidence words in NVRAM.
 - Read and verify that location 1FFF0_{hex} is 5555_{hex}.
 - Read and verify that location 1FFF2_{hex} is AAAA_{hex}.
 - Read and verify that location 1FFF4_{hex} is 1234_{hex}.
 - Read and verify that location 1FFF6_{hex} is BCDE_{hex}.

Error Index T1331 One or more of the confidence words were not correct.

Caveats The confidence words are initialized by leaving diagnostics. So, if the instrument has never entered normal operating mode, this test will fail. Once initialized, the confidence words should remain good as long as the RAM socket batteries are good.

Routine Name Internal

Overview This test verifies the **Programmable Interrupt Controller U440 <17>** by testing its registers and by generating internal interrupts.

Description

1. Save current Interrupt Controller registers.
2. Write the pattern `0005hex` to the DMA 0 Control Register, DMA 1 Control Register, Timer Control Register, Interrupt 0 Control Register, Interrupt 1 Control Register, Interrupt 2 Control Register, Interrupt 3 Control Register, Priority Mask Register, and the Interrupt Status Register.
3. Read all the previous registers and verify that each contains the written pattern.
4. Repeat steps 2 and 3 using the pattern `000Ahex` for all registers except the last two. In those cases, use the pattern `0002hex`.
5. Set the priority for each internal interrupt.
 - Write `0003hex` to DMA 0 Control Register, `0004hex` to DMA 1 Control Register, `0005hex` to Timer Control Register, and `00FDhex` to the Mask Register.
6. Write `0000hex` to the Priority Mask Register.
7. Set up DMA 0 to generate an interrupt by writing `FF07hex` to DMA 0 Control Word Register for the first pass, DMA 1 to generate an interrupt by writing `FF07hex` to DMA 1 Control Word Register for the second pass, and Timer 2 to generate an interrupt by writing `C000hex` to Timer 2 Mode/Control Register for the third pass.
8. Read and verify that the Interrupt Request Register shows a pending interrupt for DMA 0.
9. Set the Mask Register to allow the interrupt by writing `00FBhex` to the Mask Register.
10. Read and verify that the In-Service Register and Poll Status Register show no DMA 0 interrupt.
11. Set the Priority Mask Register to `0003hex` to allow DMA 0 interrupt for the first pass, to `0004` for the second pass, and `0005hex` for third pass.
12. Read and verify that the Poll Status Register shows a DMA 0 interrupt.
13. Verify that the DMA 0 interrupt handler shows that the DMA 0 interrupt was serviced.

14. Read and verify that the In-Service Register and Interrupt Request Register show no DMA 0 interrupt.
15. Set the Mask Register to 00FD_{hex}.
16. Repeat steps 7 through 15, replacing DMA 0 with DMA 1 on the second pass and with Timer 2 on the third pass.

Error Index T1411 The Transfer Count Register was not zero.

Error Index T1412 One of the interrupt registers did not contain the correct status pattern. The display shows the first pattern that failed.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name DMA0

Overview This test verifies Programmable DMA Unit's U440 <17> channel 0 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Set 35 destination memory locations to 0000_{hex}.
2. Program DMA channel 0 to transfer 35 source patterns to the destination memory.
 - Write B725_{hex} and 0023_{hex} to DMA 0 Channel Control Word and Transfer Count registers, respectively. The address values written to the Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, and Source Pointer Lower registers are determined at test execution time. Write B727_{hex} to the DMA 0 Channel Control Word to initiate the DMA transfer.
3. Wait 1 ms for DMA 0 to finish the transfer.
4. Read and verify that the Transfer Count Register is zero.
5. Read and verify the destination memory locations.

Error Index T1511 The Transfer Count Register was not zero.

Error Index T1512 One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name DMA1

Overview This test verifies **Programmable DMA Unit's** U440 <17> channel 1 by transferring a set of patterns from one group of memory locations (source) to another (destination).

Description

1. Set the 35 destination memory locations to 0000_{hex}.
2. Program the DMA channel 1 to transfer the 35 source patterns to the destination memory.
 - Write B725_{hex} and 0023_{hex} to DMA 1 Channel Control Word and Transfer Count registers, respectively. The address values written to the Destination Pointer Upper, Destination Pointer Lower, Source Pointer Upper, and Source Pointer Lower registers are determined at test execution time. Write B727_{hex} to the DMA 1 Channel Control Word to initiate the DMA transfer.
3. Wait 1 ms for DMA 1 to finish the transfer.
4. Read and verify that the Transfer Count Register is zero.
5. Read and verify the destination memory locations.

Error Index T1521 The Transfer Count Register was not zero.

Error Index T1522 One or more of the transferred patterns did not match the expected value. The display shows the first pattern that failed.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name Control

Overview This test verifies the MMU Mode Register U651 <19>. This is done by using a "walking ones" test.

Description

1. Save the current state of the MMU Mode Register U651 <19>.
2. Perform a "walking ones" test on the MMU Mode Register.
 - Write 01hex to the MMU Mode Register.
 - Read the MMU Mode Register and verify that the data read back is 01hex.
 - Continue this write/read/verify sequence with the patterns 02hex, 04hex, 08hex, 10hex, 20hex, 40hex, 80hex.
3. Restore the previous state of the MMU Mode Register.

Error Index T1611 The value read back did not match what was written.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name Handshake

Overview This test verifies **TBC-MMU Handshake Control (Input/Output)** <19>, **TBC-MMU Next DMA Request FF (Flip-Flop)** <19>, **MMU Status Read Buffer** <19>, and partially checks **MMU Bidirectional Data Port** <19>. This is done by setting up the DMA channels to transfer a message and simulating the handshake signals from the Executive Processor.

Description

1. Clear any possible message requests by pulsing **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) and **TBC-MMU Next DMA Request FF** MMU_DRGLR(L), and reading the **MMU Bidirectional Data Port**.
2. Set up CPU 80186 DMA 0 <17> to receive a one word message and DMA 1 to send a one word message.
 - Load both DMA's Source Pointer and Destination Pointer Registers to point to a one word area of memory (actual location is not important), both DMA's Transfer Count Registers with 0001_{hex}, and both DMA's Control Word Registers with 6E67_{hex}.
3. Read **MMU Status Read Buffer** U851A to verify that **TBC-MMU Next DMA Request FF** U480A-5 DRQ1 is low and **MMU Bidirectional Data Port** U680-11 DRQ0 is low. Verify that DMA 0 Transfer Count Register is still 1, and DMA 1 Transfer Count Register is still 1.
4. Start a receive message operation by pulsing **TBC-MMU Handshake Control (Input)** MMU_AOUT(L) to simulate DIGLATCH(L) from the Executive Processor.
5. Read **MMU Status Read Buffer** to verify that the send DMA request (DRQ1) is low and that the receive DMA request (DRQ0) is high. Verify that DMA 1 Transfer Count Register is still 1 and that DMA 0 Transfer Count Register is now 0.
6. Clear the receive DMA request (DRQ0) by reading the **MMU Bidirectional Data Port**, through the **MMU Message Data Transceivers**.
7. Read **MMU Status Read Buffer** to verify that the send DMA request (DRQ1) is still low and that the receive DMA request (DRQ0) is now also low. Verify that DMA 1 Transfer Count Register is still 1 and that DMA 0 Transfer Count Register is still 0.
8. Start a send message operation by writing to the **MMU Bidirectional Data Port**, through the **MMU Message Data Transceivers**.
9. Read **MMU Status Read Buffer** to verify that there are no send or receive DMA requests present (i.e. DRQ1 and DRQ0 low). Verify that DMA 1 Transfer Count Register is still 1 and that DMA 0 Transfer Count Register is still 0.

Timebase

10. Pulse **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) to simulate DOUT(L) from the Executive Processor.
11. Read **MMU Status Read Buffer** to verify that there are now send and receive DMA requests present (i.e. DRQ1 and DRQ0 high). Verify that DMA 1 Transfer Count Register is now 0 and that DMA 0 Transfer Count Register is still 0.
12. Clear the send DMA request (DRQ1) by pulsing **TBC-MMU Next DMA Request FF U491C-10 MMU_DRGLR(L)**.
13. Read **MMU Status Read Buffer** to verify that the send DMA request (DRQ1) is now low and that the receive DMA request (DRQ0) is still high. Verify that DMA 1 Transfer Count Register is still 0 and that DMA 0 Transfer Count Register is still 0.
14. Repeat the above steps 256 times.

Error Index T1621

The state of the receive control signal (DRQ0) was not correct. The expected field contains the expected control signal status. The actual field contains the DMA 0 transfer count.

Error Index T1622

The state of the send control signal (DRQ1) was not correct. The expected field contains the expected control signal status. The actual field contains the DMA 1 transfer count.

Caveats

This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name Messages

Overview This test verifies the **MMU Bidirectional Data Port** <19>, through the **MMU Message Data Transceivers** <19>. "Walking ones" patterns are written to the data port and read back by simulating the Executive Processor handshake.

Description

1. Write 0001_{hex} to the data port latches U583 & U680 <19>, through the data transceivers U561 & U661 <19>.
2. Pulse **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) <19> to strobe the test pattern back into the data port latches.
3. Read the pattern from the data port latches, through the data transceivers, and verify that it is 0001_{hex}.
4. Repeat above steps using the patterns 0002_{hex}, 0004_{hex}, 0008_{hex}, 0010_{hex}, 0020_{hex}, 0040_{hex}, 0080_{hex}, 0100_{hex}, 0200_{hex}, 0400_{hex}, 0800_{hex}, 1000_{hex}, 2000_{hex}, 4000_{hex}, 8000_{hex}.

Error Index T1631 The value read back did not match the value written on the low order 8 bits.

Error Index T1632 The value read back did not match the value written on the high order 8 bits.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name Waveforms

Overview This test verifies the **Even/Odd Select & Waveform Data Encoder** <19> by writing, reading, and verifying test patterns which are looped back through the **MMU Bidirectional Data Port** <19>.

Description

1. Write an even-byte test pattern to the **MMU Bidirectional Data Port** (see table below).
 - Set **MMU Mode Register** EVEN(L)/ODD(H) U651-12 <19> low.
 - Write the even-byte test pattern to waveform port address 39FFE_{hex}. This enables the data bits AD0-AD7 (low byte of the even-byte test pattern) present on **Even/Odd Select & Waveform Data Encoder** to be latched into the **MMU Bidirectional Data Port**.
2. Loop the test pattern from the **MMU Bidirectional Data Port** outputs back to its inputs.
 - Pulse **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) <19>.
3. Read the test pattern from the **MMU Bidirectional Data Port**, through the **MMU Message Data Transceivers**, and verify that it matches the expected pattern (see table below).
4. Write an odd-byte test pattern to the **MMU Bidirectional Data Port** (see Error Index section below).
 - Set **MMU Mode Register** EVEN(L)/ODD(H) U651-12 <19> high.
 - Write the odd-byte test pattern to waveform port address 39FFE_{hex}. This enables the data bits AD8-AD15 (high byte of the odd-byte test pattern) present on **Even/Odd Select & Waveform Data Encoder** to be latched into the **MMU Bidirectional Data Port**.
5. Loop the test pattern from the **MMU Bidirectional Data Port** outputs back to its inputs.
 - Pulse **TBC-MMU Handshake Control (Input)** MMU_DOUT(L) <19>.
6. Read the test pattern from the **MMU Bidirectional Data Port**, through the **MMU Message Data Transceivers**, and verify that it matches the expected pattern (see table below).
7. Repeat the above steps with the remaining test patterns. The following table shows the even-byte and odd-byte patterns written through the **Even/Odd Select & Waveform Data Encoder** and the associated pattern expected to be read back via the **MMU Bidirectional Data Port**.

<u>Even-Byte Test Pattern</u>	<u>Odd-Byte Test Pattern</u>	<u>Expected Pattern</u>
FF00 _{hex}	00FF _{hex}	8001 _{hex}
00FF _{hex}	FF00 _{hex}	7FFF _{hex}
807F _{hex}	7F80 _{hex}	FF00 _{hex}
FE01 _{hex}	01FE _{hex}	8100 _{hex}
FD02 _{hex}	02FD _{hex}	8200 _{hex}
FB04 _{hex}	04FB _{hex}	8400 _{hex}
F708 _{hex}	08F7 _{hex}	8800 _{hex}
EF10 _{hex}	10EF _{hex}	9000 _{hex}
DF20 _{hex}	20DF _{hex}	A000 _{hex}
BF40 _{hex}	40BF _{hex}	C000 _{hex}
7F80 _{hex}	807F _{hex}	0000 _{hex}

Error Index T1641 The even/odd bytes (i.e. AD0-AD7/AD8-AD15) were swapped. The test pattern for the odd-byte was returned when the MMU Mode Register was set for even-byte; or the test pattern for the even-byte was returned when the MMU Mode Register was set for odd-byte

Error Index T1642 The results read for the even-byte test pattern did not match the expected results.

Error Index T1643 The results read for the odd-byte test pattern did not match the expected results.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name Addressing

Overview This test verifies the MMU Write Offset Address Register <19> by using a "walking zeros" test and simulating the Executive Processor handshake.

Description

1. Write $FFFF_{hex}$ to address $39FFE_{hex}$. This latches $FFFF_{hex}$ into the MMU Bidirectional Data Port <19> (which is unimportant) and $1FFE_{hex}$ into the MMU Write Offset Address Register <19>.
2. Pulse MMU_AOUT(L) to latch the address from the MMU Write Offset Address Register into the MMU Bidirectional Data Port.
3. Read the address pattern from the MMU Bidirectional Data Port, through the MMU Message Data Transceivers, and verify that it is $1FFE_{hex}$.
4. Repeat above steps writing $FFFF_{hex}$ to the following addresses and verifying the associated expected pattern from the data port latches.

<u>Address</u>	<u>Expected Pattern</u>
$39FFD_{hex}$	$1FFD_{hex}$
$39FFB_{hex}$	$1FFB_{hex}$
$39FF7_{hex}$	$1FF7_{hex}$
$39FEF_{hex}$	$1FEF_{hex}$
$39FDF_{hex}$	$1FDF_{hex}$
$39FBF_{hex}$	$1FBF_{hex}$
$39F7F_{hex}$	$1F7F_{hex}$
$39EFF_{hex}$	$1EFF_{hex}$
$39DFF_{hex}$	$1DFF_{hex}$
$39BFF_{hex}$	$1BFF_{hex}$
$397FF_{hex}$	$17FF_{hex}$
$38FFF_{hex}$	$0FFF_{hex}$

Error Index T1651 The data read back on the low byte did not match what was expected.

Error Index T1652 The data read back on the high byte did not match what was expected.

Caveats This test requires a software protocol to take place with the Executive processor in which the Executive processor sets the MMU/Timebase busses in a tri-state mode. If the Executive processor can not accomplish this successfully, then the test will fail.

Routine Name Timer 2

Overview This test verifies **Programmable Timers U440 <17>** timer 2's counting accuracy by counting the system clock for a short duration.

Description 1. Stop the timer 2 by writing `4000hex` to its Timer Mode/Control Register.

 2. Program the counter to count up by writing `0000hex`, `FFFFhex`, and `C000hex` to timer 2 Count Register, Max Count Value A Register, and Timer 2 Mode/Control Register, respectively.

 3. Perform a software delay.

 4. Stop the timer by writing `4000hex` to its Timer Mode/Control Register.

 5. Read and verify the timer 2 count against the known value for the expected tolerance.

Error Index T1711 The timer 2 count was not within the expected tolerance.

Routine Name Timer 0

Overview This test verifies the ability of **Programmable Timers U440 <17>** timer 0 and **Reference Clock & Frequency/Period Measurement Logic <18>** to count software generated strobes.

Description

1. Set up timebase to count strobe events.
 - Set **Amode Register U850, U751 <18>** to `0003hex` to set **PDG Control Logic U1140-17 MODE0 <21>** high. Set **Tgmode Register U920, U1021 <23>** to `0002hex` to set **Reference Clock & Frequency/Period Measurement Logic U160-1 TGMODE1 <18>** high. This selects PCLK(H) as the clock input to U361B.
2. Set up Timer 0 to count strobes.
 - Load timer 0 Max A Register with 8, Max B Register with 8, Counter Register with 0, and Mode/Control Word Register with `C005hex`.
3. Verify that Timer 0 is waiting for strobes.
 - Read Timer 0 Mode/Control Word and verify that the Maximum Count (bit 5) and Register in Use (bit 14) are cleared (i.e. low or a zero).
 - Read Timer 0 Count Register and verify it is zero.
4. Generate 8 software strobes.
 - Write to address `30040hex` 8 times, with any data value, to pulse **PDG Control Logic U1140-19 PCLK(L) <21>** 8 times.
5. Verify that Timer 0 has switched from Max A Register to Max B Register and has recorded all strobes.
 - Read Timer 0 Mode/Control Word Register and verify that Maximum Count (bit 5) and Register in Use (bit 12) are set (i.e. high or a one).
 - Read Timer 0 Counter Register and verify that it is zero.
6. Generate 8 software strobes.
 - Write to address `30040hex` 8 times, with any data value, to pulse **PDG Control Logic U1140-19 PCLK(L) <21>** 8 times.
7. Verify that Timer 0 has switched from Max B Register to Max A Register and has recorded all strobes.
 - Read Timer 0 Mode/Control Word Register and verify that Maximum Count (bit 5) is set and Register in Use (bit 12) is cleared.

- Read Timer 0 Counter Register and verify that it is zero.

Error Index T1721

Timer 0 Max A Register did not correctly recognize the first 8 strobes pulses. Bits 12 and 5 of the displayed values reflect the important bits referenced above. Bits 15, 2, and 1 should always be set. If any of these bits are not set, CPU U440 <17> is most likely faulty. Any other bits can be ignored.

Error Index T1722

Timer 0 Max B Register did not correctly recognize the second 8 strobes pulses. Bits 12 and 5 of the displayed values reflect the important bits referenced above. Bits 15, 2, and 1 should always be set. If any of these bits are not set, CPU U440 <17> is most likely faulty. Any other bits can be ignored.

Routine Name Timer 1

Overview This test verifies that **Programmable Timers U440 <17>** timer 1 and **Reference Clock and Frequency/Period Measurement Logic <18>** can generate and count measurement gates correctly. Timer 1 is set up to count for 1 ms on both the Max A Register and Max B Register.

Description 1. Initialize frequency count mode.

- Set **Amode Register U850, U751 <18>** to 0003_{hex} to set **PDG Control Logic U1140-17 MODE0 <21>** high. Set **Tgmode Register U920, U1021 <23>** to 0002_{hex} to set **Reference Clock & Frequency/Period Measurement Logic U160-1 & U370A-2 TGMODE1 <18>** high. This selects PCLK(H) as the clock input to U360A.
- Load **Programmable Timers U440 <17>** Timer 0 Max A Register with 1, Max B Register with 1, Counter Register with 0, and Mode/Control Word Register with C004_{hex}. This sets **Reference Clock & Frequency/Period Measurement Logic U360A-2 BACQGAT(L)** high.
- Write to address 30040_{hex} with any data value to pulse U160-13 PCLK(L) <18>. This sets U370A-1 DLYGATE(L) high and inhibits counters U390 & U380 from counting.

2. Set up **Programmable Timers U440 <17>** Timer 1 to count.

- Load Timer 1 Max A Register with 030C_{hex}, Max B Register with 030C_{hex}, Counter Register with 0, and Mode/Control Word Register with C006_{hex}.

3. Enable counting and gate generation.

- Reset **Reference Clock & Frequency/Period Measurement Logic <18>** by reading 30028_{hex}. This causes U380-11 RESCON(H) to toggle and clear the outputs of U380.
- Set U370A-2 TGMODE1 low to allow counters U390 & U380 to run, thereby producing gate pulses on U370B-6 TMRIN1.

4. Delay for 1 ms.

5. Verify that Timer 1 has switched from Max A Register to Max B Register.

- Read Timer 1 Mode/Control Word Register and verify that Maximum Count (bit 5) and Register in Use (bit 12) are set.

6. Delay for 1 ms.

7. Verify that Timer 1 has switched from Max B Register to Max A Register.

- Read Timer 1 Mode/Control Word Register and verify that Maximum Count (bit 5) is set and Register in Use (bit 12) is cleared.

Error Index T1731

Timer 1 Max A Register did not correctly count the 1 ms time period. Bits 12 & 5 of the displayed values reflect the important bits referenced above. Bits 15, 2, and 1 should always be set. If any of these bits are not set, CPU U440 <17> is most likely faulty. Any other bits can be ignored.

Error Index T1732

Timer 1 Max B Register did not correctly count the 1 ms time period. Bits 12 & 5 of the displayed values reflect the important bits referenced above. Bits 15, 2, and 1 should always bit set. If any of these bits are not set, CPU U440 <17> is most likely faulty. Any other bits can be ignored.

Routine Name Control

Overview This test verifies **VCO Coarse Frequency Control <23>** and **VCO Fine Frequency Control <23>**. The DACs in these two areas are tested by using a "walking ones" test. VCO <23> frequency control is tested by measuring the response to various DAC settings.

- Description**
1. Verify the functionality of **VCO Coarse Frequency Control <23>** and **VCO Fine Frequency Control <23>**.
 - Set **VCO U1330-4 TGMODE4** low to connect the **VCO Coarse Frequency Control DAC** outputs to the **VCO Vcon** input.
 - Set **VCO Coarse Frequency Control U1220 DAC A** to test pattern *01hex* and **DAC B** to *00hex*.
 - Set **Diagnostic A/D Select & Convert <18> MODE4**, **MODE5**, **MODE6**, and **MODE7** to 0, 0, 0, and 0, respectively, to select **U841-13 TCOMP** as the input to **U750**.
 - Start and read **Diagnostic A/D Select & Convert** eight times.
 - Set **Diagnostic A/D Select & Convert <18> MODE4**, **MODE5**, **MODE6**, and **MODE7** to 1, 0, 1, and 0, respectively, to select **U841-5 MVCO** as the input to **U750**.
 - Start and read **Diagnostic A/D Select & Convert** eight times.
 - Set **VCO Coarse Frequency Control U1220 DAC A** to *00hex* and **DAC B** to test pattern *01hex*.
 - Start and read **Diagnostic A/D Select & Convert** eight times.
 - Set **Diagnostic A/D Select & Convert <18> MODE4**, **MODE5**, **MODE6**, and **MODE7** to 0, 1, 1, and 0, respectively, to select **U841-2 LVCO** as the input to **U750**.
 - Shift current test pattern left four bits (i.e. *01hex* becomes *10hex* on first pass) and write it to **VCO Fine Frequency Control DAC U1121 <23>**.
 - Start and read **Diagnostic A/D Select & Convert <18>** eight times.
 - Repeat the above steps for the test patterns *02hex*, *04hex*, *08hex*, *10hex*, *20hex*, *40hex*, *80hex*.
 2. Verify that all of the values read for each test pattern on the **VCO Coarse Frequency Control** and the **VCO Fine Frequency Control DACs** were within desired limits.

3. Measure and verify VCO Vcon <23>.
 - Set **Diagnostic A/D Select & Convert** <18> MODE4, MODE5, MODE6, and MODE7 to 0, 0, 0, and 0, respectively, to select U841-13 TCOMP as the input to U750.
 - Set **VCO U1330-4 TGMODE4** high to disconnect the **VCO Coarse Frequency Control** DAC outputs from the VCO Vcon input
 - Start and read **Diagnostic A/D Select & Convert** <18> eight times. This becomes the basis for the following reference value VCO Vref <23>.
 - Set **Diagnostic A/D Select & Convert** <18> MODE4, MODE5, MODE6, and MODE7 to 1, 0, 1, and 0, respectively, to select U841-5 MVCO as the input to U750.
 - Set **VCO U1330-4 TGMODE4** low to connect the **VCO Coarse Frequency Control** DAC outputs to the VCO Vcon input
 - Set **VCO Coarse Frequency Control** DAC A to 00_{hex} and DAC B to the value $2.56 * (-0.007 * Vref + 3.342) / 27$.
 - Start and read **Diagnostic A/D Select & Convert** <18> eight times.
 - Verify that this last value calculated for Vcon is within $\pm 4\%$ of the value calculated for Vref.
 - Set **VCO Coarse Frequency Control** DAC A to FF_{hex} and DAC B to the value $0.261 * (-0.007 * Vref + 3.342) - 415.23$.
 - Start and read **Diagnostic A/D Select & Convert** <18> eight times.
 - Verify that this last value calculated for Vcon is again within $\pm 4\%$ of the value calculated for Vref.
4. Find the minimum frequency reference using **VCO Fine Frequency Control** <23>.
 - Write 000_{hex} to **VCO Fine Frequency Control** DAC U1121.
 - Write 6020_{hex} to **Tgmode Register** <23>. This sets **VCO U1330-4 TGMODE4** low to connect the **VCO Coarse Frequency Control** DAC outputs to the VCO Vcon input. It also sets up **Programmable Delay Generator (Fast) U1450 GATED COUNTER** <23> to use FRQGATE(L) as a gate for clock input CLK1 on pins 67 & 68 (from **VCO U1430 COUT/COUTB**). This allows **Slow Frequency Counter VCLK** <23> to drive **Programmable Timers U440-20 TMRIN0** <17> via **Reference Clock & Frequency/Period Measurement Logic U160** <18>.

Timebase

- Write 0003_{hex} to **Amode Register <18>** to set **PDG Control Logic and Programmable Delay Generator (Slow) <21>** into parallel diagnostics mode.
 - Load **Programmable Timers <17>** timer 1 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
 - Load **Programmable Timers <17>** timer 0 Max A Register with FFFF_{hex}, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
 - Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, **Programmable Delay Generator (Fast) <23>**, and **Slow Frequency Counter <23>**.
 - Write C005_{hex} to Timer 0 Mode/Control Word Register.
 - Wait 1 ms for the VCO to stabilize.
 - Write C006_{hex} to Timer 1 Mode/Control Word Register to start the VCO frequency measurement operation. This causes **Reference Clock & Frequency/Period Measurement Logic U360B-12 BTMROUT1 and FRQGATE(L) <18>** to eventually go low, allowing the clock input of **Programmable Delay Generator (Fast) GATED COUNTER <23>** to drive **Slow Frequency Counter VCLK <23>**, which is essentially counted via **Programmable Timers TMRIN0 <17>**.
 - Wait 2 ms to allow the frequency measurement to stop via FRQGATE(L) going back high.
 - Verify that **Programmable Timers Timer 1** did terminate properly to make a valid VCO frequency measurement. If so, retain the count in timer 0 as the minimum frequency.
5. Find the maximum frequency reference using **VCO Fine Frequency Control <23>**.
 - Perform the same steps as in step 4 to find the maximum frequency reference, writing FFF_{hex} to **VCO Fine Frequency Control <23>** instead of 000_{hex}.
 6. Calculate and verify that the mid-point of the frequency measurements is within the acceptable range.
 7. Verify that the total frequency range is within the acceptable range.

Error Index T2111

The output of **VCO Coarse Frequency Control U1220 DAC B <23>** was not within the expected range.

Timebase

Strobe Gen

VCO

Control (T211X)

Error Index T2112	The output of VCO Coarse Frequency Control U1220 DAC A <23> was not within the expected range.
Error Index T2113	The output of VCO Fine Frequency Control U1121 <23> was not within the expected range.
Error Index T2114	The VCO mid-point frequency was not within the expected range.
Error Index T2115	The total VCO frequency range, using the VCO Fine Frequency Control DAC U1121 <23> at minimum and maximum settings, was not within the expected range.
Error Index T2116	The temperature compensation could not be adjusted within the expected range (i.e. Vcon vs. Vref).
Error Index T2117	The VCO frequency could not be measured.

Timebase

Strobe Gen

VCO

Ramp DACS ()

Routine Name Ramp DACS

Overview This test provides stimulus for the three DACS that control the VCO <23> frequency. The resulting waveform is a sawtooth from each DAC.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector set to "Off". When this test is invoked, the operator may use an oscilloscope to view the resultant sawtooth waveforms.

Description

1. Sequentially increment **VCO Course Frequency Control U1220 DAC B <23>** through all DAC codes (i.e. *00hex* through *FFhex*).
2. Sequentially increment **VCO Course Frequency Control U1220 DAC A <23>** through all DAC codes (i.e. *00hex* through *FFhex*).
3. Sequentially increment **VCO Fine Frequency Control DAC U1121 <23>** through all DAC codes in steps of *10hex* (i.e. *000hex*, *010hex*, *020hex*, ..., to *FF0hex*).

Error Index None.

Timebase

Routine Name	Control (Control Register)
Overview	This test verifies Amode Register U850 and U751 <18> by performing a "walking one's" test.
Description	<ol style="list-style-type: none">1. Perform a "walking ones" test on Amode Register. Terminate test if any verify operation fails.<ul style="list-style-type: none">• Write the pattern 0001<i>hex</i> to address 3001<i>Chex</i>. Read the same address and verify that it was 0001<i>hex</i>. Continue this write/read/verify sequence with the patterns 0002<i>hex</i>, 0004<i>hex</i>, 0008<i>hex</i>, 0010<i>hex</i>, 0020<i>hex</i>, 0040<i>hex</i>, 0080<i>hex</i>, 0100<i>hex</i>, 0200<i>hex</i>, 0400<i>hex</i>, 0800<i>hex</i>, 1000<i>hex</i>, 2000<i>hex</i>, 4000<i>hex</i>, 8000<i>hex</i>.
Error Index T2211	The value read back did not match what was written.

Routine Name

Pipeline

Overview

This test verifies **Correction Pipeline Register** U1160, U1161, U1061, and U1060 <21> and partially verifies **PDG Control Logic** <21>. This is done by serially shifting different "walking one's" patterns into each correction pipeline shadow register (contained internally in each device), transferring the shadow register to the parallel register, transferring the parallel register back to the shadow register, and then serially reading the shadow register.

Description

1. Perform a "walking one's" test on the **Correction Pipeline Register** <21>, using pattern *01hex* in U1161 (FINE), *04hex* in U1160 (SLOPE), *10hex* in U1061 (POSITION - low), and *40hex* in U1060 (POSITION - high). Terminate test if any verify operation fails.
 - Set **PDG Control Logic** MODE0 <21> and **Correction Pipeline Register** MODE3 low.
 - Clock the test patterns into U1161 (BD1), U1160 (BD2), U1061 (BD3), and U1060 (BD4) by pulsing **Correction Pipeline Register** DDCLK(H) <21> eight times to load the full 8-bit patterns into each device. This is done by writing the appropriate data bits to address *3004Ahex* each time.
 - Set **PDG Control Logic** <21> MODE0 and **Correction Pipeline Register** MODE3 high.
 - Pulse **Correction Pipeline Register** PCLK(H) <21> to transfer the shadow registers to the pipeline registers of each device by writing to *30040hex*.
 - Pulse DDCLK(H) to transfer the pipeline registers to the shadow registers of each device.
 - Set **Correction Pipeline Register** MODE3 low.
 - Read the test patterns from U1161 (SDO1), U1160 (SDO2), U1061 (SDO3), and U1060 (SDO4) by pulsing DDCLK(H) <21> eight times. This is done by reading the data bits from address *3004Ahex* each time.
 - Verify that the pattern read back matches what was written.
2. Repeat the write/read/verify sequence of step 1 with the following patterns (passes 2-8) for each device:

Timebase

Strobe Gen

Strbe Digitl

Pipeline (T222X)

	Pass 1	Pass 2	Pass 3	Pass 4	Pass 5	Pass 6	Pass 7	Pass 8
U1161	01hex	02hex	04hex	08hex	10hex	20hex	40hex	80hex
U1160	04hex	08hex	10hex	20hex	40hex	80hex	01hex	02hex
U1061	10hex	20hex	40hex	80hex	01hex	020hex	04hex	08hex
U1060	40hex	80hex	01hex	02hex	04hex	08hex	10hex	20hex

Error Index T2221 The pattern read back from **Correction Pipeline Register U1161 (FINE)** <21> did not match what was written.

Error Index T2222 The pattern read back from **Correction Pipeline Register U1160 (SLOPE)** <21> did not match what was written.

Error Index T2223 The pattern read back from **Correction Pipeline Register U1061 (POSITION - low)** <21> did not match what was written.

Error Index T2224 The pattern read back from **Correction Pipeline Register U1060 (POSITION - high)** <21> did not match what was written.

Timebase

Routine Name Slow Count

Overview This test verifies **Programmable Delay Generator (slow)** U1250 <21>, **Parallel To Serial Converter** U930 and U940 <21>, and partially verifies **PDG Control Logic** U1140 and U1040 <21> by shifting various patterns through the registers in **Programmable Delay Generator (slow)** U1250 <21>.

- Description**
1. Verify the addressability and accessibility of various registers in **Programmable Delay Generator (slow)** U1250 <21>.
 - Set **PDG Control Logic** MODE0 high (to allow processor control of the U1140 clocks), **Programmable Delay Generator (slow)** MODE1 low (to select serial diagnostics mode), and MODE2 low (to disable the counters in U1250 from counting).
 - For each register given in Table I below, set **PDG Control Logic** MODE8 and MODE9 to the given levels and use the specified clock to transfer the associated pattern into the register. This is done by shifting each bit, via **PDG Control Logic** BD0, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).
 - Read and verify the pattern in each register by serial shifting the test pattern out of the register through **Programmable Delay Generator (slow)** and **Serial Read Buffer** SER0. This is done by pulsing the appropriate clock signal 24 or 48 times and reading BD0 of **Serial Read Buffer** each time.
 - Repeat the load/read/verify sequence for the sets of register patterns in Table II and III.

TABLE I

U1250 Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	11222335566	ACLK	x	x
HOLD OFF COUNTER	556677	HCLK	x	x
STROBE DELAY COUNTER	667788	YCLK	x	x
STROBE DELAY SHADOW	445566778899	BCLK	0	1
HOLD OFF DELAY SHADOW	889900	HHCLK	1	0
DOT DELAY SHADOW	667788990011	DCLK	0	0

TABLE II

U1250 Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	555555555555	ACLK	x	x
HOLD OFF COUNTER	555555	HCLK	x	x
STROBE DELAY COUNTER	555555	YCLK	x	x
STROBE DELAY SHADOW	555555555555	BCLK	0	1
HOLD OFF DELAY SHADOW	555555	HHCLK	1	0
DOT DELAY SHADOW	555555555555	DCLK	0	0

Timebase

TABLE III

U1250 Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	AAAAAAAAAAAA	ACLK	x	x
HOLDOFF COUNTER	AAAAAA	HCLK	x	x
STROBE DELAY COUNTER	AAAAAA	YCLK	x	x
STROBE DELAY SHADOW	AAAAAAAAAAAA	BCLK	0	1
HOLDOFF DELAY SHADOW	AAAAAA	HHCLK	1	0
DOT DELAY SHADOW	AAAAAAAAAAAA	DCLK	0	0

2. Verify that the shadow registers in **Programmable Delay Generator (slow)** U1250 <21> will transfer to the appropriate counter registers.
 - Load the shadow registers given in table I below using the method described in step 1.
 - Pulse **Programmable Delay Generator (slow)** HOF_LEAST(H), HLD(H), and YLD(H).
 - Read and verify the registers given in table II below using the method described in step 1.

TABLE I

U1250 Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY SHADOW	112233445566	BCLK	0	1
HOLDOFF DELAY SHADOW	556677	HHCLK	1	0
DOT DELAY SHADOW	334455667788	DCLK	0	0

TABLE II

U1250 Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	112233445566	ACLK	x	x
HOLDOFF COUNTER	556677	HCLK	x	x
STROBE DELAY COUNTER	112233	YCLK	x	x

3. Verify the functionality of **Programmable Delay Generator (slow)** <21> ACLK(H) and U1250 FULL ADDER.
 - Load the STROBE DELAY SHADOW REGISTER and DOT DELAY SHADOW REGISTER with 555555555555_{hex}, using the method described in step 1.
 - Pulse HOF_LEAST(H) to load the 48 BIT STROBE DELAY and 48 BIT DOT DELAY REGISTERs with the contents of their shadow registers.
 - Set MODE1 high (to select normal mode), and MODE2 high (to enable the counters in U1250 to count).

- Pulse ACLK(H) to add the 48 BIT DOT DELAY REGISTER to the 48 BIT STROBE DELAY REGISTER.
 - Set MODE1 low (to select serial diagnostics mode), and MODE2 low (to disable the counters in U1250 from counting).
 - Read and verify that the 48 BIT STROBE DELAY REGISTER is `AAAAAAAAAAAAhex`, using the method described in step 1
 - Repeat the previous items in this step, loading the registers with `AAAAAAAAAAAAhex` and checking for `55555555554hex`.
4. Verify the functionality of **Programmable Delay Generator (slow)** <21> YCLK(H) and U1250 STROBE DELAY COUNTER by having each counter stage ripple up.
- Load the STROBE DELAY COUNTER with the first test pattern from the table below, using the method describe in step 1.
 - Set MODE1 high (to select normal mode), and MODE2 high (to enable the counters in U1250 to count).
 - Pulse YCLK(H) to make the counter increment by one.
 - Set MODE1 low (to select serial diagnostics mode), and MODE2 low (to disable the counters in U1250 from counting).
 - Read and verify that the STROBE DELAY COUNTER contains the expected pattern from the table below, using the method describe in step 1.
 - Repeat the previous items in this step for the rest of the patterns in the following table.

Test Pattern (hex)	Expected Pattern (hex)
000001	000002
000003	000004
000007	000008
00000F	000010
00001F	000020
00003F	000040
00007F	000080
0000FF	000100
0001FF	000200
0003FF	000400
0007FF	000800
000FFF	001000
001FFF	002000
003FFF	004000
007FFF	008000
00FFFF	010000
01FFFF	020000
03FFFF	040000
07FFFF	080000
0FFFFF	100000
1FFFFF	200000
3FFFFF	400000
7FFFFF	800000
FFFFFF	000000

5. Verify the functionality of **Programmable Delay Generator (slow)** <21> HCLK(H) and U1250 24 BIT HOLDOFF COUNTER by having each counter stage ripple up.
 - Repeat step 4, using HCLK(H) instead of YCLK(H) to test the 24 BIT HOLDOFF COUNTER.
6. Verify the terminal count output (SDTTLB(H)) of the **Programmable Delay Generator (slow)** <21> STROBE DELAY COUNTER.
 - Load the STROBE DELAY COUNTER with FFFFFFFE_{hex}, using the method describe in step 1.
 - Verify that **Serial Read Buffer U1041** <21> bit 5 is low.
 - Set MODE1 high (to select normal mode), and MODE2 high (to enable the counters in U1250 to count).
 - Pulse YCLK(H) to make the counter increment by one.
 - Set MODE1 low (to select serial diagnostics mode), and MODE2 low (to disable the counters in U1250 from counting).
 - Verify that **Serial Read Buffer U1041** <21> bit 5 is high.

7. Verify the terminal count output (SDTTLBP(H)) of the **Programmable Delay Generator (slow)** <21> 24 BIT HOLDOFF COUNTER.
 - Load the 24 BIT HOLDOFF COUNTER with FFFFE_{hex}, using the method describe in step 1.
 - Verify that **Serial Read Buffer U1041** <21> bit 6 is low.
 - Set MODE1 high (to select normal mode), and MODE2 high (to enable the counters in U1250 to count).
 - Pulse HCLK(H) to make the counter increment by one.
 - Set MODE1 low (to select serial diagnostics mode), and MODE2 low (to disable the counters in U1250 from counting).
 - Verify that **Serial Read Buffer U1041** <21> bit 6 is high.
8. Verify that the **Parallel To Serial Converter** <21> loads the shadow registers of **Programmable Delay Generator (slow)** U1250 <21> properly.
 - Set **Programmable Delay Generator (slow)** MODE1 high (to select normal mode).
 - For each of the registers in the following table, set **PDG Control Logic** MODE8 and MODE9 to the values given, load the **Parallel To Serial Converter** twice with the test pattern, and then allow **Programmable Timers** <17> Timer 1 (via **PDG Control Logic** U1040-16 GCLK <21>) to clock the data into the appropriate shadow register through SDOUT. Verify that the timer counter reached zero each time.

U1250 Register	Parallel To Serial Converter Test		Clock	MODE9	MODE8
	Pattern (hex)				
HOLDOFF DELAY SHADOW	5555	HHCLK	1	0	
STROBE DELAY SHADOW	55AA	BCLK	0	1	
DOT DELAY SHADOW	AAAA	DCLK	0	0	

- Read and verify that the **HOLDOFF DELAY SHADOW REGISTER** contains 005555_{hex}, using the method described in step 1.
- Read and verify that the **STROBE DELAY SHADOW REGISTER** contains 0000000055AA_{hex}, using the method described in step 1.
- Read and verify that the **DOT DELAY SHADOW REGISTER** contains 00000000AAAA_{hex}, using the method described in step 1.

Error Index T2231

The 48 BIT STROBE DELAY REGISTER of **Programmable Delay Generator (slow)** U1250 could not be accessed.

Strobe Gen

Strbe Digitl

Slow Count (T223X)

Error Index T2232	The 24 BIT HOLDOFF COUNTER of Programmable Delay Generator (slow) U1250 could not be accessed.
Error Index T2233	The STROBE DELAY COUNTER of Programmable Delay Generator (slow) U1250 could not be accessed.
Error Index T2234	The STROBE DELAY SHADOW REGISTER of Programmable Delay Generator (slow) U1250 could not be accessed.
Error Index T2235	The HOLDOFF DELAY SHADOW REGISTER of Programmable Delay Generator (slow) U1250 could not be accessed.
Error Index T2236	The DOT DELAY SHADOW REGISTER of Programmable Delay Generator (slow) U1250 could not be accessed.
Error Index T2237	The STROBE DELAY SHADOW REGISTER of Programmable Delay Generator (slow) U1250 could not be transferred to the 48 BIT STROBE DELAY REGISTER.
Error Index T2238	The HOLDOFF DELAY SHADOW REGISTER of Programmable Delay Generator (slow) U1250 could not be transferred through the 24 BIT HOLDOFF DELAY REGISTER to the 24 BIT HOLDOFF COUNTER.
Error Index T2239	The upper 24 bits of the 48 BIT STROBE DELAY REGISTER of Programmable Delay Generator (slow) U1250 could not be loaded into the STROBE DELAY COUNTER.
Error Index T223A	The 48 BIT STROBE DELAY REGISTER of Programmable Delay Generator (slow) U1250 and the 48 BIT DOT DELAY REGISTER did not add correctly.
Error Index T223B	The STROBE DELAY COUNTER of Programmable Delay Generator (slow) U1250 did not count up correctly.
Error Index T223C	The 24 BIT HOLDOFF COUNTER of Programmable Delay Generator (slow) U1250 did not count up correctly.
Error Index T223D	The STROBE DELAY COUNTER terminal count (SDTTLB) of Programmable Delay Generator (slow) U1250 <21> did not behave correctly.
Error Index T223E	The 24 BIT HOLDOFF COUNTER terminal count (SDTTLP) of Programmable Delay Generator (slow) U1250 <21> did not behave correctly.
Error Index T223F	Programmable Timers <17> Timer 1 did not generate the correct number of GCLK <21> pulses or the Parallel To Serial Converter <21> did not correctly shift test patterns into one or more of the Programmable Delay Generator (slow) <21> shadow registers (through PDG Control Logic <21>).

Timebase

Routine Name Corct Table (Correction Table)

Overview This test verifies the address and data lines to **Slope LUT** (Lookup Table) <21> and **Position LUT** <21> by performing a "walking one's" test on the data lines and a RAM test on the lookup table memories.

Description

- Set the **Slope LUT** and **Position LUT** <21> address to zero.
 - For each **Programmable Delay Generator (slow)** register given in the following table, set **PDG Control Logic** MODE8 and MODE9 to the given levels and use the specified clock to transfer zero into the register. This is done by shifting each bit, via **PDG Control Logic** BD0, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Register	Clock	MODE9	MODE8
STROBE DELAY SHADOW	BCLK	0	1
HOLD OFF DELAY SHADOW	HHCLK	1	0
DOT DELAY SHADOW	DCLK	0	0

 - Pulse **Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages.
 - Set **Correction Pipeline Register** MODE3 <21> low.
- Write the first test pattern from the table below to **Strobe LUT** <21> and **Position LUT** <21>. Bits 0 - 3 go to U1151 and U1050, bits 4 - 7 go to U1150 and U951, and bits 8 - 11 go to U950.
- Transfer the **Strobe LUT** <21> and **Position LUT** <21> test pattern to **Correction Pipeline Register** <21>.
 - Set **PDG Control Logic** <21> MODE0 high (to allow processor control of the U1140 clocks).
 - Pulse PCLK(H).
- Read and verify the test pattern from **Correction Pipeline Register** <21>.
 - Set **Correction Pipeline Register** MODE3 high.
 - Pulse DDCLK(H) to transfer the pipeline registers to their internal shadow registers.
 - Set **Correction Pipeline Register** MODE3 low.
 - Serially read the test results by pulsing DDCLK(H) eight times, accumulating the SDO2, SDO3, and SDO4 data bits from **Correction Pipeline Register** via **Serial Read Buffer** <21>.

Timebase

- Verify that the pattern read matches the test pattern written (from the following table).

Test Pattern (hex)	Expected Slope LUT Pattern (hex)	Expected Position LUT Pattern (hex)
001	01	001
002	02	002
004	04	004
008	08	008
010	10	010
020	20	020
040	40	040
080	80	080
100	00	100
200	00	200
400	00	400
800	00	800

- Repeat steps 2 - 4 for the remaining test patterns in the table.
5. Initialize Slope LUT and Position LUT <21> RAM address range 00hex to FFhex with 55hex and 555hex, respectively.
 - Load Programmable Delay Generator (slow) <21> U1250 STROBE DELAY SHADOW REGISTER, HOLDOFF DELAY SHADOW REGISTER, and DOT DELAY SHADOW REGISTER with 000000000000hex, 000000hex, 0000000000400hex, respectively, using the method described in step 1. This will allow lookup table addresses SD10 - SD17 to be incremented from 00hex through FFhex, via the 48 BIT STROBE DELAY REGISTER.
 - Write 55hex to Slope LUT address 00hex and 555hex to Position LUT address 00hex.
 - Set PDG Control Logic MODE0 high (to allow processor control of the U1140 clocks), Programmable Delay Generator (slow) MODE1 high (to select parallel diagnostics mode), and MODE2 high (to enable the counters in U1250).
 - Pulse Programmable Delay Generator (slow) ACLK(H) to effectively increment the lookup table address.
 - Repeat the previous three items 255 more times to load all the RAM locations in both lookup tables.
 6. Disable address incrementing.
 - Load Programmable Delay Generator (slow) U1250 STROBE DELAY SHADOW REGISTER, HOLDOFF DELAY SHADOW

Timebase

REGISTER, and DOT DELAY SHADOW REGISTER with zero, using the method described in step 1.

7. Verify **Slope LUT** and **Position LUT** RAM <21> address range 00_{hex} to FF_{hex} .
 - Set **Programmable Delay Generator (slow)** MODE1 low (to select serial diagnostics mode).
 - Serial shift the lookup table's address (i.e. 000000000000_{hex} on the first pass) into the STROBE DELAY REGISTER (via SDOUT) by pulsing ACLK(H) 48 times.
 - Set **Programmable Delay Generator (slow)** MODE1 high (to select parallel diagnostics mode).
 - Pulse PCLK(H) to transfer the data at address 00_{hex} of both lookup tables into the pipeline register.
 - Perform the same procedure as in step 4 above to read the **Correction Pipeline Register**.
 - Verify that the data read from the SLOPE pipeline register is 55_{hex} and the data from the POSITION pipeline register is 555_{hex} .
 - If both the SLOPE and POSITION pipeline registers verified correctly, then write AA_{hex} to **Slope LUT** address 00_{hex} and AA_{hex} to **Position LUT** address 00_{hex} .
 - Increment the **Programmable Delay Generator (slow)** U1250 48 BIT STROBE DELAY REGISTER by 400_{hex} . This effectively increments the lookup table address by one.
 - Repeat the previous items in this step until all values in the lookup tables have been checked for $55_{hex}/555_{hex}$ and replaced with AA_{hex}/AA_{hex} (i.e. 255 more times).
8. Repeat step 7, again starting at lookup table address 00_{hex} , checking for AA_{hex}/AA_{hex} and filling with $00_{hex}/000_{hex}$ in **Slope LUT** and **Position LUT** <21>, respectively.

Error Index T2241

The data bits of **Slope LUT** U1151 did not match one of the expected patterns during the data lines test.

Error Index T2242

The data bits of **Slope LUT** U1150 did not match one of the expected patterns during the data lines test.

Error Index T2243

The data bits of **Position LUT** U1050 did not match one of the expected patterns during the data lines test.

Error Index T2244

The data bits of **Position LUT** U951 did not match one of the expected patterns during the data lines test.

Timebase

Strobe Gen

Strbe Digitl

Corct Table (T224X)

Error Index T2245	The data bits of Position LUT U950 did not match one of the expected patterns during the data lines test.
Error Index T2246	The pattern read from the displayed memory location in Slope LUT U1151 was not the pattern written.
Error Index T2247	The pattern read from the displayed memory location in Slope LUT U1150 was not the pattern written.
Error Index T2248	The pattern read from the displayed memory location in Position LUT U1050 was not the pattern written.
Error Index T2249	The pattern read from the displayed memory location in Position LUT U951 was not the pattern written.
Error Index T224A	The pattern read from the displayed memory location in Position LUT U950 was not the pattern written.

Timebase

Routine Name SDS (Strobe Deskew for Main Delay Adjust)

Overview This test verifies **Position DAC** U1480 <21>, **Slope DAC** U1280 <21>, and **Main Delay Adjust** U1570 <23>. The DAC's are tested by loading "walking ones" values and then reading the voltages back through **Diagnostic A/D Select & Convert** <18>. **Main Delay Adjust** is checked by taking loop delay time samples every half volt step of the control signal (VDL CNTRL) over the entire range of the **Position DAC** and then applying a least-squares fit to the data.

Description

1. Perform "walking one's" test on **Slope DAC** U1280 <21> and **Position DAC** U1480 <21>.
 - Set **Diagnostic A/D Select & Convert** <18> **MODE4**, **MODE5**, **MODE6**, and **MODE7** to 0, 1, 0, and 0, respectively, to select U841-15 **ANSLOP** as the input to diagnostic A/D U750.
 - Set **PDG Control Logic** **MODE0** <21> high (to allow processor control of the U1140 clocks) and **Programmable Delay Generator (slow)** **MODE1** low (to select serial diagnostics mode).
 - Set **Correction Pipeline Register** **MODE3** <21> low.
 - Write 01_{hex} to **Slope LUT** <21> U1151 and U1150 and then pulse **Correction Pipeline Register** **PCLK(H)** to latch the value into **SLOPE** register U1160.
 - Start and read **Diagnostic A/D Select & Convert** eight times. Average the values and save for later verification.
 - Set **Diagnostic A/D Select & Convert** <18> **MODE4**, **MODE5**, **MODE6**, and **MODE7** to 1, 1, 0, and 0, respectively, to select U841-12 **ANDLY** as the input to diagnostic A/D U750.
 - For each **Programmable Delay Generator (slow)** <21> register given in the following table, set **PDG Control Logic** **MODE8** and **MODE9** to the given levels and use the specified clock to transfer zero into the register. This is done by shifting each bit, via **PDG Control Logic** **BD0**, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Register	Clock	MODE9	MODE8
STROBE DELAY SHADOW	BCLK	0	1
HOLD OFF DELAY SHADOW	HHCLK	1	0
DOT DELAY SHADOW	DCLK	0	0

- Pulse **Programmable Delay Generator (slow)** **HOF_LEAST(H)** to transfer the internal shadow registers of U1250 into the next register stages.
- Write 00_{hex} to **Slope LUT** <21> and 010_{hex} to **Position LUT**.

Timebase

- Pulse **Correction Pipeline Register** PCLK(H) to latch the values of the lookup tables into the SLOPE and POSITION registers of the pipeline register.
 - Start and read **Diagnostic A/D Select & Convert** eight times. Average the values and save for later verification.
 - Repeat the previous items of this step, using the patterns 02hex, 04hex, 08hex, 10hex, 20hex, 40hex, and 80hex in the first part to test the **Slope DAC** and the patterns 020hex, 040hex, 080hex, 100hex, 200hex, 400hex, and 800hex in the second part to test the **Position DAC**.
 - Verify that all of the averaged results are within expected ranges.
2. Set the **VCO <23>** to its default values.
- Set **VCO Coarse Frequency Control <23>** DAC part a to 80hex and DAC part b to B4hex.
 - Set **VCO Fine Frequency Control <23>** to 800hex.
 - Set **Delay Adjust DAC <23>** to 800hex.
3. Set the delay control (**VDL CNTRL**) of **Main Delay Adjust U1570 <23>**.
- Write the first test pattern from the following table to **Position LUT <21>**.

Test Pattern (hex)

FF0
EF0
DF0
CF0
BF0
AF0
9F0
8F0
7F0
6F0
5F0
4F0
3F0
2F0
1F0
0F0

- Pulse **Correction Pipeline Register** PCLK(H) to latch the test pattern in **Position LUT** into the POSITION registers of the pipeline register. This sets **Position DAC VDL CNTRL** to the desired value.

Timebase

4. Measure the VCO frequency.

- Write `6020hex` to **Tgmode Register <23>**. This sets **VCO U1330-4 TGMODE4** low to connect the **VCO Coarse Frequency Control DAC** outputs to the **VCO Vcon** input. It also sets up **Programmable Delay Generator (fast) U1450 GATED COUNTER <23>** to use **FRQGATE(L)** as a gate for clock input **CLK1** on pins 67 & 68 (from **VCO U1430 COUT/COUTB**). This allows **Slow Frequency Counter VCLK <23>** to drive **Programmable Timers U440-20 TMRIN0 <17>** via **Reference Clock & Frequency/Period Measurement Logic U160 <18>**.
- Write `0003hex` to **Amode Register <18>** to set **PDG Control Logic** and **Programmable Delay Generator (slow) <21>** into parallel diagnostics mode.
- Load **Programmable Timers <17>** Timer 1 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with `4000hex`.
- Load **Programmable Timers <17>** Timer 0 Max A Register with `FFFFhex`, Counter Register with 0, and Mode/Control Word Register with `4000hex`.
- Pulse **RESCON(L)** to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, **Programmable Delay Generator (fast) <23>**, and **Slow Frequency Counter <23>**.
- Write `C005hex` to Timer 0 Mode/Control Word Register.
- Wait 1 ms for the VCO to stabilize.
- Write `C006hex` to Timer 1 Mode/Control Word Register to start the VCO frequency measurement operation. This causes **Reference Clock & Frequency/Period Measurement Logic U360B-12 BTMROUT1** and **FRQGATE(L) <18>** to eventually go low, allowing the clock input of **Programmable Delay Generator (fast) GATED COUNTER <23>** to drive **Slow Frequency Counter VCLK <23>**, which is essentially counted via **Programmable Timers TMRIN0 <17>**.
- Wait 2 ms to allow the frequency measurement to stop via **FRQGATE(L)** going back high.
- Verify that **Programmable Timers Timer 1 <17>** terminated properly in order to make a valid VCO frequency measurement. If so, retain the count in Timer 0.
- Repeat the previous items of this step seven more times, average the results, and save this VCO frequency value for the following loop delay time calculations.

Timebase

5. Measure the loop delay time with **Programmable Delay Generator Generator (fast)** STR <23> (i.e. the strobe delay) as the delay feedback element.
- Set **Trigger Level DAC** U800 <22> to FF_{hex} (-1 volt).
 - Write 60A2_{hex} to **Tgmode Register** <23> and 5387_{hex} to **Amode Register** <18> to stop strobes and set up timebase.
 - Load **Programmable Timers** <17> Timer 0 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
 - Load **Programmable Timers** <17> Timer 1 Max A Register with FFFF_{hex}, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
 - Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic** <18>, **Programmable Delay Generator (fast)** <23>, and **Slow Frequency Counter** <23>.
 - For each **Programmable Delay Generator (slow)** register given in the following table, set **PDG Control Logic** MODE8 and MODE9 to the given levels and use the specified clock to transfer the given initial pattern into the register. This is done by shifting each bit, via **PDG Control Logic** BD0, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFFA900000	BCLK	0	1
HOLD OFF DELAY	FFFFFFE	HHCLK	1	0
DOT DELAY	000000000000	DCLK	0	0

- Pulse **Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages. This also sets the least 6 bits of the overall 30-bit holdoff value to an initial value of 29_{hex} by loading A4_{hex} into **Holdoff Register** U941 <23>.
- Pulse **Programmable Delay Generator (slow)** HLD and YLD to transfer the **HOLD OFF DELAY** register into the **HOLD OFF COUNTER** and the **STROBE DELAY** register into the **STROBE DELAY COUNTER**.
- Enable **Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.

Timebase

- Write *8386hex* to **Amode Register** <18> to select the strobe as the delay feedback element (via **Variable Holdoff** MFSTR <22>) and to set the timebase into normal mode of operation (mode bits 0 & 1).
 - Pulse **Variable Holdoff** TG_ARM(L) <22> to allow triggering to occur correctly.
 - Write *60A3hex* to **Tgmode Register** <23> to enable freerun mode in the **Trigger Recognizer** <22>.
 - Write *C005hex* to **Timer 1 Mode/Control Word Register** and *C006hex* to **Timer 0 Mode/Control Word Register** to start the delay measurement operation.
 - Wait approximately 3 ms.
 - Verify that **Programmable Timers** **Timer 0** <17> terminated properly in order to make a valid delay measurement. If so, use the previously saved VCO frequency measurement in step 4 to calculate the actual loop delay time. Retain this calculated loop delay value.
 - Repeat the previous items of this step seven more times, average the results, and save for use in step 7.
6. Repeat steps 3, 4, and 5 for the remaining test patterns in the table of step 3.
 7. Perform a least-squares fit of the collected loop delay times accumulated in step 5 and verify that the calculated least-squares error and slope are within expected ranges.

Error Index T2311	The value read back from Position DAC U1480 <21>, through Diagnostic A/D Select & Convert <18>, was not within the expected range.
Error Index T2312	The value read back from Slope DAC U1280 <21>, through Diagnostic A/D Select & Convert <18>, was not within the expected range.
Error Index T2313	The calculated least-squares error of the loop delay times of Main Delay Adjust <23> was not within the expected range.
Error Index T2314	The calculated least-squares slope of the loop delay times of Main Delay Adjust <23> was not within the expected range.
Error Index T2315	The VCO frequency could not be measured.
Error Index T2316	The loop delay time could not be measured.

Timebase

Routine Name Fast Count

Overview This test verifies the delay generators (for strobe and holdoff delays) in **Programmable Delay Generator (fast) U1450 <23>**. This is done by measuring the loop delay times when using U1450 STR and STRP as the delay feedback elements.

Description

1. Set the VCO <23> to its default values.
 - Set VCO Coarse Frequency Control <23> DAC part a to 80hex and DAC part b to B4hex.
 - Set VCO Fine Frequency Control <23> to 800hex.
 - Set Delay Adjust DAC <23> to 800hex.
2. Measure the VCO frequency.
 - Write 6020hex to Tgmode Register <23>. This sets VCO U1330-4 TGMODE4 low to connect the VCO Coarse Frequency Control DAC outputs to the VCO Vcon input. It also sets up Programmable Delay Generator (fast) U1450 GATED COUNTER <23> to use FRQGATE(L) as a gate for clock input CLK1 on pins 67 & 68 (from VCO U1430 COUT/COUTB). This allows Slow Frequency Counter VCLK <23> to drive Programmable Timers U440-20 TMRIN0 <17> via Reference Clock & Frequency/Period Measurement Logic U160 <18>.
 - Write 0003hex to Amode Register <18> to set PDG Control Logic and Programmable Delay Generator (slow) <21> into parallel diagnostics mode.
 - Load Programmable Timers <17> Timer 1 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000hex.
 - Load Programmable Timers <17> Timer 0 Max A Register with FFFFhex, Counter Register with 0, and Mode/Control Word Register with 4000hex.
 - Pulse RESCON(L) to reset Reference Clock & Frequency/Period Measurement Logic <18>, Programmable Delay Generator (fast) <23>, and Slow Frequency Counter <23>.
 - Write C005hex to Timer 0 Mode/Control Word Register.
 - Wait 1 ms for the VCO to stabilize.
 - Write C006hex to Timer 1 Mode/Control Word Register to start the VCO frequency measurement operation. This causes Reference Clock & Frequency/Period Measurement Logic U360B-12 BTMROUT1 and FRQGATE(L) <18> to eventually go low, allowing

the clock input of **Programmable Delay Generator (fast) GATED COUNTER <23>** to drive **Slow Frequency Counter VCLK <23>**, which is essentially counted via **Programmable Timers TMRIN0 <17>**.

- Wait 2 ms to allow the frequency measurement to stop via FRQGATE(L) going back high.
 - Verify that **Programmable Timers Timer 1 <17>** terminated properly in order to make a valid VCO frequency measurement. If so, retain the count in Timer 0.
 - Repeat the previous items of this step seven more times, average the results, and save this VCO frequency value for the following loop delay time calculations.
3. Measure the loop delay time with **Programmable Delay Generator (fast) STRP <23>** (i.e. the holdoff delay) as the delay feedback element.
- Set **Trigger Level DAC U800 <22>** to FF_{hex} (-1 volt).
 - Write $60A2_{hex}$ to **Tgmode Register <23>** and 5387_{hex} to **Amode Register <18>** to stop strobes and set up timebase.
 - Load **Programmable Timers <17>** Timer 0 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
 - Load **Programmable Timers <17>** Timer 1 Max A Register with $FFFF_{hex}$, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
 - Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, **Programmable Delay Generator (fast) <23>**, and **Slow Frequency Counter <23>**.
 - For each **Programmable Delay Generator (slow)** register given in the following table, set **PDG Control Logic MODE8** and **MODE9** to the given levels and use the specified clock to transfer the given initial pattern into the register. This is done by shifting each bit, via **PDG Control Logic BD0**, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFEA40000	BCLK	0	1
HOLDOFF DELAY	FFFFFA	HHCLK	1	0
DOT DELAY	000000000000	DCLK	0	0

Timebase

- **Pulse Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages. This also sets the least 6 bits of the overall 30-bit holdoff value to an initial value of 24_{hex} by loading 90_{hex} into **Holdoff Register** U941 <23>.
 - **Pulse Programmable Delay Generator (slow)** HLD and YLD to transfer the HOLDOFF DELAY register into the HOLDOFF COUNTER and the STROBE DELAY register into the STROBE DELAY COUNTER.
 - **Enable Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.
 - Write 4386_{hex} to **Amode Register** <18> to select the holdoff as the delay feedback element (via **Variable Holdoff STRP** <22>) and to set the timebase into normal mode of operation (mode bits 0 & 1).
 - **Pulse Variable Holdoff** TG_ARM(L) <22> to allow triggering to occur correctly.
 - Write 60A3_{hex} to **Tgmode Register** <23> to enable freerun mode in the **Trigger Recognizer** <22>.
 - Write C005_{hex} to **Timer 1 Mode/Control Word Register** and C006_{hex} to **Timer 0 Mode/Control Word Register** to start the delay measurement operation.
 - Wait approximately 3 ms.
 - Verify that **Programmable Timers** **Timer 0** <17> terminated properly in order to make a valid delay measurement. If so, use the previously saved VCO frequency measurement in step 2 to calculate the actual loop delay time. Retain this calculated loop delay value.
 - Repeat the previous items of this step seven more times, average the results, and verify that this value is within the expected range.
4. Repeat step 2 to get another VCO frequency measurement.
 5. Measure the loop delay time with **Programmable Delay Generator (fast)** STR <23> (i.e. the strobe delay) as the delay feedback element.
 - Set **Trigger Level DAC** U800 <22> to FF_{hex} (-1 volt).
 - Write 60A2_{hex} to **Tgmode Register** <23> and 5387_{hex} to **Amode Register** <18> to stop strobes and set up timebase.

Timebase

- Load **Programmable Timers** <17> Timer 0 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
- Load **Programmable Timers** <17> Timer 1 Max A Register with FFFF_{hex}, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
- Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic** <18>, **Programmable Delay Generator (fast)** <23>, and **Slow Frequency Counter** <23>.
- For each **Programmable Delay Generator (slow)** register given in the following table, set PDG Control Logic MODE8 and MODE9 to the given levels and use the specified clock to transfer the given initial pattern into the register. This is done by shifting each bit, via **PDG Control Logic BD0**, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFFA900000	BCLK	0	1
HOLDOFF DELAY	FFFFFFE	HHCLK	1	0
DOT DELAY	000000000000	DCLK	0	0

- Pulse **Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages. This also sets the least 6 bits of the overall 30-bit holdoff value to an initial value of 29_{hex} by loading A4_{hex} into **Holdoff Register U941** <23>.
- Pulse **Programmable Delay Generator (slow)** HLD and YLD to transfer the HOLDOFF DELAY register into the HOLDOFF COUNTER and the STROBE DELAY register into the STROBE DELAY COUNTER.
- Enable **Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.
- Write 8386_{hex} to **Amode Register** <18> to select the strobe as the delay feedback element (via **Variable Holdoff MFSTR** <22>) and to set the timebase into normal mode of operation (mode bits 0 & 1).
- Pulse **Variable Holdoff** TG_ARM(L) <22> to allow triggering to occur correctly.
- Write 60A3_{hex} to **Tgmode Register** <23> to enable freerun mode in the **Trigger Recognizer** <22>.

Timebase

- Write C005_{hex} to Timer 1 Mode/Control Word Register and C006_{hex} to Timer 0 Mode/Control Word Register to start the delay measurement operation.
- Wait approximately 3 ms.
- Verify that **Programmable Timers** Timer 0 <17> terminated properly in order to make a valid delay measurement. If so, use the previously saved VCO frequency measurement in step 4 to calculate the actual loop delay time. Retain this calculated loop delay value.
- Repeat the previous items of this step seven more times, average the results, and verify that this value is within the expected range.

Error Index T2321 The loop delay time was not within the expected range when **Programmable Delay Generator (fast)** U1450 STRP <23> (i.e. the holdoff delay) was used as the delay feedback element.

Error Index T2322 The loop delay time was not within the expected range when **Programmable Delay Generator (fast)** U1450 STR <23> (i.e. the strobe delay) was used as the delay feedback element.

Error Index T2323 The VCO frequency could not be measured.

Error Index T2324 The loop delay time could not be measured.

Routine Name Holdoff

Overview This test verifies the **Variable Holdoff <22>** circuitry. The loop delay time is measured when the variable holdoff is set at both ends of its programmable range, via DAC U1230, and the difference in the loop delay times is verified.

Description

1. Set the **VCO <23>** to its default values.
 - Set **VCO Coarse Frequency Control <23>** DAC part a to *80hex* and DAC part b to *B4hex*.
 - Set **VCO Fine Frequency Control <23>** to *800hex*.
 - Set **Delay Adjust DAC <23>** to *800hex*.
2. Set **Variable Holdoff <22>** to minimum.
 - Write *FFhex* to DAC U1230 <22>.
3. Measure the minimum variable holdoff loop delay time with **Programmable Delay Generator Generator (fast) STRP <23>** (i.e. the holdoff delay) as the delay feedback element.
 - Set **Trigger Level DAC U800 <22>** to *FFhex* (-1 volts).
 - Write *60A2hex* to **Tgmode Register <23>** and *5387hex* to **Amode Register <18>** to stop strobes and set up timebase.
 - Load **Programmable Timers <17>** Timer 0 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with *4000hex*.
 - Load **Programmable Timers <17>** Timer 1 Max A Register with *FFFFhex*, Counter Register with 0, and Mode/Control Word Register with *4000hex*.
 - Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, **Programmable Delay Generator (fast) <23>**, and **Slow Frequency Counter <23>**.
 - For each **Programmable Delay Generator (slow)** register given in the following table, set **PDG Control Logic MODE8** and **MODE9** to the given levels and use the specified clock to transfer the given initial pattern into the register. This is done by shifting each bit, via **PDG Control Logic BD0**, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFEA40000	BCLK	0	1
HOLDOFF DELAY	FFFFFFA	HHCLK	1	0
DOT DELAY	000000000000	DCLK	0	0

- **Pulse Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages. This also sets the least 6 bits of the overall 30-bit holdoff value to an initial value of 24_{hex} by loading 90_{hex} into **Holdoff Register** U941 <23>.
 - **Pulse Programmable Delay Generator (slow)** HLD and YLD to transfer the HOLDOFF DELAY register into the HOLDOFF COUNTER and the STROBE DELAY register into the STROBE DELAY COUNTER.
 - **Enable Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.
 - Write 4386_{hex} to **Amode Register** <18> to select the holdoff as the delay feedback element (via **Variable Holdoff** STRP <22>) and to set the timebase into normal mode of operation (mode bits 0 & 1).
 - **Pulse Variable Holdoff** TG_ARM(L) <22> to allow triggering to occur correctly.
 - Write $60A3_{hex}$ to **Tgmode Register** <23> to enable freerun mode in the **Trigger Recognizer** <22>.
 - Write $C005_{hex}$ to **Timer 1 Mode/Control Word Register** and $C006_{hex}$ to **Timer 0 Mode/Control Word Register** to start the delay measurement operation.
 - Wait approximately 3 ms.
 - Verify that **Programmable Timers** **Timer 0** <17> terminated properly in order to make a valid delay measurement. If so, save the 16-bit **Programmable Timers** **Timer 1 Count Register** <17> and the 8-bits read from **Reference Clock & Frequency/Period Measurement Logic** U381 <18> as the minimum variable holdoff loop delay (with the 8-bits from U381 being the lsb's of the result).
4. Set **Variable Holdoff** <22> to maximum.
 - Write 00_{hex} to **DAC** U1230 <22>.
 5. Repeat step 3 to find the maximum variable holdoff loop delay time.
 6. Verify that the difference between the maximum and minimum loop delay times is within the expected range.

Timebase

Strobe Gen

Strbe Analog

Holdoff (T233X)

Error Index T2331 The difference between the maximum and minimum loop delay times, caused by varying **Variable Holdoff** DAC U1230 <22>, was not within the expected range. Each count of the results fields represents 5 ps of delay.

Error Index T2332 The loop delay time could not be measured.

Timebase

Routine Name Ref Clocks (Reference Clocks)

Overview This test verifies the relationship between **Reference Clock & Frequency/Period Measurement Logic Y280** (200 MHz oscillator) <18> and **CPU Y430** (16 MHz crystal).

The test first configures **Internal Clock Generator <16B>** (which runs off the processor clock CLKOUT) to generate 1 MHz triggers into the **VCO <23>** (via **Trig Source Selector & Amplifier TRIG SIG <22>** and **Trigger Recognizer <22>**). This in turn drives **Programmable Delay Generator (fast) U1450 STRTTL <23>**, which ends up controlling **PDG Control Logic PCLK <21>**. **Reference Clock & Frequency/Period Measurement Logic PCLK <18>** clocks **Programmable Timers U440 <17>** Timer 0, which essentially counts 4096 triggers. While Timer 0 is counting triggers, **Reference Clock & Frequency/Period Measurement Logic <18>** and **Programmable Timers U440 <17>** Timer 1 count time, based on the 200 MHz oscillator Y280. At the end of the 4096 triggers, the accumulated time is checked.

The second part of the test sets the **Internal Clock Generator <16B>** to run at 100 KHz and checks the timing in the same manner as just described.

Description

1. Set internal clock for 1 MHz.
 - Write $FFFE_{hex}$ to **Internal Clock Generator <16B>** and wait 100 ms for the internal clock to reload.
2. Measure and verify the amount of time required for 4096 triggers.
 - Set **Trigger Level DAC U800 <22>** to $4C_{hex}$ (0.4 volts).
 - Write $60C2_{hex}$ to **Tgmode Register <23>** and 5387_{hex} to **Amode Register <18>** to stop strobes and set up timebase.
 - Load **Programmable Timers <17>** Timer 0 Max A Register with 10, Max B Register with 4096, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
 - Load **Programmable Timers <17>** Timer 1 Max A Register with $FFFF_{hex}$, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
 - Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, and **Programmable Delay Generator (fast) <23>**.
 - For each **Programmable Delay Generator (slow)** register given in the following table, set **PDG Control Logic MODE8** and **MODE9** to the given levels and use the specified clock to transfer the given pattern into the register. This is done by shifting each bit, via **PDG Control Logic BD0**, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

Timebase

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFF000000	BCLK	0	1
HOLDOFF DELAY	FFFFFFC	HHCLK	1	0
DOT DELAY	000000000000	DCLK	0	0

- **Pulse Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages.
 - **Pulse Programmable Delay Generator (slow)** HLD and YLD to transfer the HOLDOFF DELAY register into the HOLDOFF COUNTER and the STROBE DELAY register into the STROBE DELAY COUNTER.
 - **Enable Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.
 - Write 4386_{hex} to **Amode Register** <18> to set the timebase for normal mode of operation (mode bits 0 & 1).
 - **Pulse Variable Holdoff** TG_ARM(L) <22> to allow triggering to occur correctly.
 - Write 60C2_{hex} to **Tgmode Register** <23>.
 - Write C005_{hex} to **Timer 1 Mode/Control Word Register** and C006_{hex} to **Timer 0 Mode/Control Word Register** to start the timing measurement.
 - Wait approximately 5 ms.
 - Verify that **Programmable Timers** Timer 0 <17> terminated properly in order to make a valid timing measurement. If so, verify that the accumulated time read from **Programmable Timers** <21> **Timer 1 Counter Register** (the time measurement high byte) and **Reference Clock & Frequency/Period Measurement Logic** U381 <18> (the time measurement low byte) is within the expected limits.
3. Repeat steps 1 and 2, setting the internal clock to 100 KHz instead of 1 MHz by writing FFEC_{hex} to **Internal Clock Generator** <16B>.

Error Index T2341

Programmable Timers Timer 0 <17> did not terminate its counting sequence properly, therefore a valid timing measurement could not be made.

Error Index T2342

The timing measurement of the reference clocks was not within the expected range when the internal clock rate was set to 1 MHz.

Error Index T2343

The timing measurement of the reference clocks was not within the expected range when the internal clock rate was set to 100 KHz.

Timebase

Routine Name	SDS Plot (Strobe Deskew linearity plot for Main Delay Adjust or Between Head Delay Adjust)
Overview	This routine plots the Main Delay Adjust <23> or Between Head Delay Adjust <15> (<7> for SM-11 Multi-Channel Units) linearity curves. The routine measures and plots the loop delay times across the entire range of the selected delay device.
Operator Procedure	This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off". Once this test is invoked, the operator is prompted for relevant test information.
Description	<ol style="list-style-type: none"> 1. Ask the user whether he wants the plot for Main Delay Adjust <23> or the plot for Between Head Delay Adjust <15> (<7> for SM-11's). <p>For Main Delay Adjust <23></p> <ol style="list-style-type: none"> 2. Ask the user if he wants continuous update of the plot. <p>For Between Head Delay Adjust <15> (<7> for SM-11's)</p> <ol style="list-style-type: none"> 2. Determine which channel to plot. <ul style="list-style-type: none"> • For 11801 only, ask the user for the MCU or Mainframe identification. • Ask the user for the channel. • Ask the user whether he wants the plot the coarse DAC or the fine DAC for the channel. • Ask the user if he wants continuous update of the plot. 3. Clear the display and print the plot selection, graticule, and axis labels. 4. Set the VCO <23> to its default values. <ul style="list-style-type: none"> • Set VCO Coarse Frequency Control <23> DAC part a to 80_{hex} and DAC part b to B4_{hex}. • Set VCO Fine Frequency Control <23> to 800_{hex}. • Set Delay Adjust DAC <23> to 800_{hex}. 5. Set the reference delay.

For Main Delay Adjust <23>

- Write the pattern `FF0hex` to **Position LUT** <21>.
- Pulse **Correction Pipeline Register** PCLK(H) <21> to transfer the lookup table value set in the last item to the correction pipeline POSITION registers (this controls **Main Delay Adjust** <23> via **Position DAC** VDL CNTRL <21>).

For Between Head Delay Adjust <15> Coarse Control

- Write the pattern `0080hex` to the **Between Head Delay Control** <16A> (<6> for SM-11's) course and fine DACs for the part of the DAC which corresponds to the channel the user entered.

For Between Head Delay Adjust <15> Fine Control

- Write the pattern `8000hex` to the **Between Head Delay Control** <16A> (<6> for SM-11's) course and fine DACs for the part of the DAC which corresponds to the channel the user entered.

6. Measure the VCO frequency.

- Write `6020hex` to **Tgmode Register** <23>. This sets **VCO** U1330-4 TGMODE4 low to connect the **VCO Coarse Frequency Control** DAC outputs to the **VCO** Vcon input. It also sets up **Programmable Delay Generator (fast)** U1450 GATED COUNTER <23> to use FRQGATE(L) as a gate for clock input CLK1 on pins 67 & 68 (from **VCO** U1430 COUT/COUTB). This allows **Slow Frequency Counter** VCLK <23> to drive **Programmable Timers** U440-20 TMRIN0 <17> via **Reference Clock & Frequency/Period Measurement Logic** U160 <18>.
- Write `0003hex` to **Amode Register** <18> to set **PDG Control Logic** and **Programmable Delay Generator (slow)** <21> into parallel diagnostics mode.
- Load **Programmable Timers** <17> **Timer 1** Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with `4000hex`.
- Load **Programmable Timers** <17> **Timer 0** Max A Register with `FFFFhex`, Counter Register with 0, and Mode/Control Word Register with `4000hex`.
- Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic** <18>, **Programmable Delay Generator (fast)** <23>, and **Slow Frequency Counter** <23>.
- Write `C005hex` to **Timer 0** Mode/Control Word Register.

Timebase

- Wait 1 ms for the VCO to stabilize.
 - Write $C006_{hex}$ to Timer 1 Mode/Control Word Register to start the VCO frequency measurement operation. This causes **Reference Clock & Frequency/Period Measurement Logic U360B-12** BTMROUT1 and FRQGATE(L) <18> to eventually go low, allowing the clock input of **Programmable Delay Generator (fast) GATED COUNTER <23>** to drive **Slow Frequency Counter VCLK <23>**, which is essentially counted via **Programmable Timers TMRIN0 <17>**.
 - Wait 2 ms to allow the frequency measurement to stop via FRQGATE(L) going back high.
 - Verify that **Programmable Timers** Timer 1 <17> terminated properly in order to make a valid VCO frequency measurement. If so, retain the count in Timer 0.
 - Repeat the previous items of this step three more times, average the results, and save this VCO frequency value for the following loop delay time calculation.
7. Measure the reference loop delay time with **Programmable Delay Generator (fast) STRP <23>** (i.e. the holdoff delay) as the delay feedback element.
- Set **Trigger Level DAC U800 <22>** to $4C_{hex}$ (0.4 volts).
 - Write $60A2_{hex}$ to **Tgmode Register <23>** and 5387_{hex} to **Amode Register <18>** to stop strobes and set up timebase.
 - Load **Programmable Timers <17>** Timer 0 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
 - Load **Programmable Timers <17>** Timer 1 Max A Register with $FFFF_{hex}$, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
 - Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, **Programmable Delay Generator (fast) <23>**, and **Slow Frequency Counter <23>**.
 - For each **Programmable Delay Generator (slow)** register given in the following table, set **PDG Control Logic MODE8** and **MODE9** to the given levels and use the specified clock to transfer the given initial pattern into the register. This is done by shifting each bit, via **PDG Control Logic BD0**, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

Timebase

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFFA900000	BCLK	0	1
HOLDOFF DELAY	FFFFFFE	HHCLK	1	0
DOT DELAY	000000000000	DCLK	0	0

- **Pulse Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages. This also sets the least 6 bits of the overall 30-bit holdoff value to an initial value of 29_{hex} by loading $A4_{hex}$ into **Holdoff Register** U941 <23>.
- **Pulse Programmable Delay Generator (slow)** HLD and YLD to transfer the HOLDOFF DELAY register into the HOLDOFF COUNTER and the STROBE DELAY register into the STROBE DELAY COUNTER.
- **Enable Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.
- Write 4386_{hex} to **Amode Register** <18> to select the holdoff as the delay feedback element (via **Variable Holdoff** STRP <22>) and to set the timebase into normal mode of operation (mode bits 0 & 1).
- Write $60A2_{hex}$ to **Tgmode Register** <23> to inhibit freerun mode in the **Trigger Recognizer** <22>.
- **Pulse Variable Holdoff** TG_ARM(L) <22> and then **Trig Source Selector & Amplifier** TG_PULSE(L) <22> to allow triggering to occur correctly.
- Write $C005_{hex}$ to **Timer 1 Mode/Control Word Register** and $C006_{hex}$ to **Timer 0 Mode/Control Word Register** to start the delay measurement operation.
- Wait approximately 3 ms.
- Verify that **Programmable Timers** Timer 0 <17> terminated properly in order to make a valid delay measurement. If so, use the previously saved VCO frequency measurement in step 6 to calculate the actual loop delay time. Retain this calculated loop delay value.
- Repeat the previous items of this step three more times, average the results, and save as the reference loop delay time.

Timebase

8. Set the delay to be measured.

For Main Delay Adjust <23>

- Write the pattern `FF0hex` to **Position LUT <21>**. On each successive pass through this step, decrement the pattern by `10hex`.
- **Pulse Correction Pipeline Register PCLK(H) <21>** to transfer the lookup table value set in the last item to the correction pipeline **POSITION** registers (this controls **Main Delay Adjust <23>** via **Position DAC VDL CNTRL <21>**).

For Between Head Delay Adjust <15> Coarse Control

- Write the pattern `0080hex` to the **Between Head Delay Control <16A>** (<6> for SM-11's) course and fine DACs for the part of the DAC which corresponds to the channel the user entered. On each successive pass through this step, increment the pattern by `100hex` (thereby exercising the course control DAC only).

For Between Head Delay Adjust <15> Fine Control

- Write the pattern `8000hex` to the **Between Head Delay Control <16A>** (<6> for SM-11's) course and fine DACs for the part of the DAC which corresponds to the channel the user entered. On each successive pass through this step, increment the pattern by 1 (thereby exercising the fine control DAC only).

9. Repeat steps 6 and 7 to find the loop delay time to be measured.
10. Use the measured loop delay time from step 9 and the reference loop delay time from step 7 to calculate and plot the actual loop delay time on the display.
11. Repeat steps 5 through 10 255 times to plot the delay curve for the selected delay device.
12. Find the minimum loop delay time.

For Main Delay Adjust <23>

- Write the pattern `FFFhex` to **Position LUT <21>**.
- **Pulse Correction Pipeline Register PCLK(H) <21>** to transfer the lookup table value set in the last item to the correction pipeline **POSITION** registers (this controls **Main Delay Adjust <23>** via **Position DAC VDL CNTRL <21>**).

Timebase

For **Between Head Delay Adjust** <15> Course or Fine Control

- Write the pattern `0000hex` to the **Between Head Delay Control** <16A> (<6> for SM-11's) course and fine DACs for the part of the DAC which corresponds to the channel the user entered.
- Repeat steps 6 and 7 (averaging eight values instead of four) and display the resulting loop delay time as the minimum loop delay time.

13. Find the maximum loop delay time.

- Repeat step 12 using `000hex` for **Main Delay Adjust** or `FFFFhex` for **Between Head Delay Adjust** Course or Fine Control and display the resulting loop delay time as the maximum loop delay time.

Error Index

None.

Routine Name Ramp DACS

Overview This routine provides stimulus for **Main Delay Adjust <23>** or **Between Head Delay Adjust <15>** (<7> for SM-11 Multi-Channel Units). For **Main Delay Adjust**, this is done by providing successive sawtooth waveforms on **Position DAC U1480 <21>**, **Fine Interpolator Multiplying DAC U1380 <21>**, and **Slope DAC U1280 <21>**. For **Between Head Delay Adjust <15>** (<7> for SM-11 Multi-Channel Units), the sawtooth waveform stimulus comes from **Between Head Delay Control <16A>** (<6> for SM-11's).

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off". Once this test is invoked, the operator is prompted for relevant test information.

Description

1. Ask the user whether he wants to generate stimulus for **Main Delay Adjust <23>** or for **Between Head Delay Adjust <15>** (<7> for SM-11's).

For **Main Delay Adjust <23>**

2. Generate the sawtooth waveforms.
 - Write *00hex* to **Fine Interpolator Multiplying DAC U1380 <21>** and **Slope DAC U1280 <21>** by serially shifting the pattern into the **Correction Pipeline Register <21>** FINE and SLOPE registers via the BD1 and BD2 lines, respectively
 - Write *000hex* through *FFFhex* in steps of 4 to **Position DAC U1480 <21>** by serially shifting the pattern into the **Correction Pipeline Register <21>** POSITION registers via the BD3 and BD4 lines. (Keep the other DACs constant by reloading them with zero each step of the position DAC.)
 - Write *FFhex* to **Fine Interpolator Multiplying DAC U1380 <21>** and *000hex* to **Position DAC U1480 <21>** as previously described.
 - Write *00hex* through *FFhex*, as previously described, to **Slope DAC U1280 <21>**, holding the fine interpolator and position DACs constant at their values from the previous item.
 - Write *FFhex* to **Slope DAC U1280 <21>** and *000hex* to **Position DAC U1480 <21>** as previously described.
 - Write *00hex* through *FFhex*, as previously described, to **Fine Interpolator Multiplying DAC U1380 <21>**, holding the slope and position DACs constant at their values from the previous item.
3. Repeat the previous step until the user stops the test.

For **Between Head Delay Adjust <15>** (<7> for SM-11's)

Timebase

2. Determine which channel to stimulate.

For 11801 only

- Ask the user for Mainframe or the MCU identification.
- Ask the user for the channel.

For 11802 only

- Ask the user for the channel.

3. Generate the sawtooth waveform.

- Write the following patterns to the **Between Head Delay Control** <16A> (<6> for SM-11's) course and fine DACs for the parts of the DACs which correspond to the channel the user entered.

0000hex
 0001hex
 0002hex
 0004hex
 0008hex
 0010hex
 0020hex
 0040hex
 0080hex
 0100hex
 0200hex
 0400hex
 0800hex
 1000hex
 2000hex
 4000hex
 8000hex

4. Repeat the previous step until the user stops the test.

Error Index

None.

Strobe Gen

Trigger

Control (T241X)

Routine Name

Control (Control Register)

Overview

This test verifies **Tgmode Register** U920 and U1021 <23> by performing a "walking one's" test.

Description

1. Perform a "walking ones" test on **Tgmode Register** <23>. Terminate test if any verify operation fails.
 - Write the pattern *0001hex* to **Tgmode Register**. Read **Tgmode Register** and verify that it was *0001hex*. Continue this write/read/verify sequence with the patterns *0002hex*, *0004hex*, *0008hex*, *0010hex*, *0020hex*, *0040hex*, *0080hex*, *0100hex*, *0200hex*, *0400hex*, *0800hex*, *1000hex*, *2000hex*, *4000hex*, *8000hex*.

Error Index T2411

The value read back did not match what was written.

Timebase

Routine Name Level

Overview

This test verifies **Trigger Level DAC U800 <22>** by loading it with "walking one's" values and then reading the voltages (TRIG LEVEL) back through **Diagnostic A/D Select & Convert <18>**. The differences in each DAC step are verified.

Description

1. Perform a "walking one's" test on **Trigger Level DAC U800 <22>**, saving the data each pass.
 - Set **Diagnostic A/D Select & Convert <18>** MODE4, MODE5, MODE6, and MODE7 to 0, 0, 1, and 0, respectively, to select U841-1 TRIG LEVEL as the input to diagnostic A/D U750.
 - Write the test pattern *01hex* to **Trigger Level DAC U800 <22>**.
 - Start and read **Diagnostic A/D Select & Convert <18>** eight times. Average the values and save for later verification.
 - Repeat this write/read sequence (last two items) using the patterns *02hex, 04hex, 08hex, 10hex, 20hex, 40hex, 80hex*.
2. Determine the difference between each two successive points saved in step 1 and verify that the last five differences are within the expected range (the first two differences may be inconclusive due to circuit noise).

Error Index T2421

The difference in successive "walking one's" steps of **Trigger Level DAC U800 <22>** was not within the expected range.

Routine Name

Dly Adj (Delay Adjust)

Overview

This test verifies **Delay Adjust DAC U1120 <23>** and partially verifies **VCO U1430 <23>**. The DAC is checked by loading it with "walking one's" values and then reading the voltages (DADJ) back through **Diagnostic A/D Select & Convert <18>**. The VCO is checked by measuring the difference in loop delay times caused by the DAC.

Description

1. Perform a "walking one's" test on **Delay Adjust DAC U1120 <23>**, saving the data each pass.
 - Set **Diagnostic A/D Select & Convert <18>** MODE4, MODE5, MODE6, and MODE7 to 1, 0, 0, and 0, respectively, to select U841-14 DADJ as the input to diagnostic A/D U750.
 - Write the test pattern 001hex to **Delay Adjust DAC U1120 <23>**.
 - Start and read **Diagnostic A/D Select & Convert <18>** eight times. Average the values and save for later verification.
 - Repeat this write/read sequence (last two items) using the patterns 002hex, 004hex, 008hex, 010hex, 020hex, 040hex, 080hex, 100hex, 200hex, 400hex, 800hex.
2. Verify that each of the DAC values from step 1 are within expected ranges, otherwise terminate the test.
3. Set the **VCO <23>** to its default values.
 - Set **VCO Coarse Frequency Control <23>** DAC part a to 80hex and DAC part b to B4hex.
 - Set **VCO Fine Frequency Control <23>** to 800hex.
4. Set the **Delay Adjust DAC U1120 <23>** to its minimum value.
 - Write C00hex to **Delay Adjust DAC U1120 <23>**
5. Measure the VCO frequency.
 - Write 6020hex to **Tgmode Register <23>**. This sets **VCO U1330-4 TGMODE4** low to connect the **VCO Coarse Frequency Control DAC** outputs to the **VCO Vcon** input. It also sets up **Programmable Delay Generator (fast) U1450 GATED COUNTER <23>** to use **FRQGATE(L)** as a gate for clock input **CLK1** on pins 67 & 68 (from **VCO U1430 COUT/COUTB**). This allows **Slow Frequency Counter VCLK <23>** to drive **Programmable Timers U440-20 TMRIN0 <17>** via **Reference Clock & Frequency/Period Measurement Logic U160 <18>**.

- Write 0003_{hex} to Amode Register <18> to set PDG Control Logic and Programmable Delay Generator (slow) <21> into parallel diagnostics mode.
 - Load Programmable Timers <17> Timer 1 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
 - Load Programmable Timers <17> Timer 0 Max A Register with FFFF_{hex}, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
 - Pulse RESCON(L) to reset Reference Clock & Frequency/Period Measurement Logic <18>, Programmable Delay Generator (fast) <23>, and Slow Frequency Counter <23>.
 - Write C005_{hex} to Timer 0 Mode/Control Word Register.
 - Wait 1 ms for the VCO to stabilize.
 - Write C006_{hex} to Timer 1 Mode/Control Word Register to start the VCO frequency measurement operation. This causes Reference Clock & Frequency/Period Measurement Logic U360B-12 BTMROUT1 and FRQGATE(L) <18> to eventually go low, allowing the clock input of Programmable Delay Generator (fast) GATED COUNTER <23> to drive Slow Frequency Counter VCLK <23>, which is essentially counted via Programmable Timers TMRIN0 <17>.
 - Wait 2 ms to allow the frequency measurement to stop via FRQGATE(L) going back high.
 - Verify that Programmable Timers Timer 1 <17> terminated properly in order to make a valid VCO frequency measurement. If so, retain the count in Timer 0.
 - Repeat the previous items of this step seven more times, average the results, and save this VCO frequency value for the following loop delay time calculation.
6. Measure the loop delay time with Programmable Delay Generator Generator (fast) STR <23> (i.e. the strobe delay) as the delay feedback element.
- Set Trigger Level DAC U800 <22> to FF_{hex} (-1 volt).
 - Write 60A2_{hex} to Tgmode Register <23> and 5387_{hex} to Amode Register <18> to stop strobes and set up timebase.
 - Load Programmable Timers <17> Timer 0 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.

- Load **Programmable Timers** <17> Timer 1 Max A Register with $FFFF_{hex}$, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
- Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic** <18>, **Programmable Delay Generator (fast)** <23>, and **Slow Frequency Counter** <23>.
- For each **Programmable Delay Generator (slow)** register given in the following table, set PDG Control Logic MODE8 and MODE9 to the given levels and use the specified clock to transfer the given initial pattern into the register. This is done by shifting each bit, via PDG Control Logic BD0, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFA900000	BCLK	0	1
HOLDOFF DELAY	FFFFFE	HHCLK	1	0
DOT DELAY	000000000000	DCLK	0	0

- Pulse **Programmable Delay Generator (slow)** HOF_LEAST(H) to transfer the internal shadow registers of U1250 into the next register stages. This also sets the least 6 bits of the overall 30-bit holdoff value to an initial value of 29_{hex} by loading $A4_{hex}$ into **Holdoff Register** U941 <23>.
- Pulse **Programmable Delay Generator (slow)** HLD and YLD to transfer the HOLDOFF DELAY register into the HOLDOFF COUNTER and the STROBE DELAY register into the STROBE DELAY COUNTER.
- Enable **Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.
- Write 8386_{hex} to **Amode Register** <18> to select the strobe as the delay feedback element (via **Variable Holdoff MFSTR** <22>) and to set the timebase into normal mode of operation (mode bits 0 & 1).
- Pulse **Variable Holdoff** TG_ARM(L) <22> to allow triggering to occur correctly.
- Write $60A3_{hex}$ to **Tgmode Register** <23> to enable freerun mode in the **Trigger Recognizer** <22>.
- Write $C005_{hex}$ to Timer 1 Mode/Control Word Register and $C006_{hex}$ to Timer 0 Mode/Control Word Register to start the delay measurement operation.
- Wait approximately 3 ms.

Timebase

- Verify that **Programmable Timers** Timer 0 <17> terminated properly in order to make a valid delay measurement. If so, use the previously saved VCO frequency measurement in step 4 to calculate the actual loop delay time. Retain this calculated loop delay value.
 - Repeat the previous items of this step seven more times, average the results, and save as the minimum loop delay time.
7. Set the **Delay Adjust DAC** U1120 <23> to its maximum value.
 - Write 400_{hex} to **Delay Adjust DAC** U1120 <23>
 - 8 Repeat steps 5 and 6 to find the maximum loop delay time.
 9. Verify that the difference between the maximum and minimum loop delay times is within the expected range.

Error Index T2431	One of the values read back from Delay Adjust DAC U1120 <23>, through Diagnostic A/D Select & Convert <18>, was not within the expected range.
Error Index T2432	The change in loop delay times, as Delay Adjust DAC U1120 <23> was varied from its minimum to maximum, was not within the expected ranged.
Error Index T2433	The VCO frequency could not be measured.
Error Index T2434	The loop delay time could not be measured.

Routine Name Calibrator

Overview This test provides stimulus for the Calibrator Assembly <4> by successively generating 50 KHz and 100 KHz square waves on the FRONT PANEL CALIBRATOR (OUTPUT) connector J92. The test leaves the 100 KHz waveform running at completion.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector set to "Off". When this test is invoked, the operator may use an oscilloscope to view the resultant square waves from the FRONT PANEL CALIBRATOR (OUTPUT).

Description

1. Set **Internal Clock Generator** <16B> to generate a 100 KHz signal. (This does not necessarily turn on the calibrator output.)
 - Write *FFEC_{hex}* to **Internal Clock Generator** <16A>.
2. Turn off the FRONT PANEL CALIBRATOR (OUTPUT) <4> for two seconds.
 - Write *02_{hex}* to **Internal Clock, TDR Drive Control Logic U510** <16B>.
 - Delay for two seconds.
3. Turn on the FRONT PANEL CALIBRATOR (OUTPUT) <4> to produce a 50 KHz signal (25% duty cycle) for two seconds.
 - Write *06_{hex}* to **Internal Clock, TDR Drive Control Logic U510** <16B>.
 - Delay for two seconds.
4. Turn on the FRONT PANEL CALIBRATOR (OUTPUT) <4> to produce a 100 KHz signal (50% duty cycle) for two seconds.
 - Write *04_{hex}* to **Internal Clock, TDR Drive Control Logic U510** <16B>.
 - Delay for two seconds.

Error Index None.

Routine Name Ramp DACS

Overview This test provides stimulus for the **Trigger Level DAC U800 <22>** and **Delay Adjust DAC U1120 <23>** by generating sawtooth waveforms from each DAC.

Operator Procedure This test requires operator interaction and may only be executed in the "Routine" menu with the "All" selector set to "Off". When this test is invoked, the operator may use an oscilloscope to view the resultant sawtooth waveforms.

Description

1. Ramp **Trigger Level DAC U800 <22>** through all DAC codes by writing *00hex* through *FFhex* to U800.
2. Ramp **Delay Adjust DAC U1120 <23>** through all DAC codes by writing *400hex* through *C00hex* to U1120.

Error Index None.

Routine Name

Strobes

Overview

This routine prompts the user for strobe-related setup information. This information is loaded into various strobe-related hardware registers when the exerciser is executed.

Description

1. Display current settings for strobe-related parameters.
 - Clear the display screen.
 - Display current Initial Dot Delay.
 - Display Dot Delay.
 - Display Holdoff.
 - Display the Number of Strobes.
 - Display the Number of Dummy Strobes.
 - Display the Holdoff Arm (the holdoff feedback element).
 - Display the Lead/Trail Switch.
2. Prompt the user for which item to change.
3. Take one of the following actions based on the users choice from the keypad.

INITIAL DOT DELAY

- Prompt the user for the high, middle, and low words (16-bit values) of **Programmable Delay Generator (slow) U1250 48 BIT STROBE DELAY REGISTER <21>**.

DOT DELAY

- Prompt the user for the high, middle, and low words (16-bit values) of **Programmable Delay Generator (slow) U1250 48 BIT DOT DELAY REGISTER <21>**.

HOLDOFF

- Prompt the user for the high and low words (16-bit values) of **Programmable Delay Generator (slow) U1250 24 BIT HOLDOFF DELAY REGISTER <21>**. (The lower 6 bits of the high word are used for the **Holdoff Register <23>**, not the holdoff delay register.)

NUMBER OF STROBES

- Prompt the user for the number of strobes (for use in **Programmable Timers U440 <17> timer 0**).

Timebase

NUMBER OF DUMMY STROBES

- Prompt the user for the number of dummy strobes (for use in **Programmable Timers** U440 <17> timer 0).

HOLDOFF ARM

- Ask the user if he wants the holdoff or strobe as the feedback to arm the holdoff (for **Variable Holdoff** <22> MODE14).

LEAD/TRAIL

- Ask the user if he wants lead or trail for the lead/trail switch (for **Strobe Lead/Tail Switch** <23> MODE12).

4. Repeat the previous steps until the user exits the menu.

Error Index

None.

See Also

The Strobe Gen Exerciser Execute test.

Routine Name Trigger

Overview This routine prompts the user for trigger-related information. This information is loaded into various trigger-related hardware registers when the exerciser is executed.

Description

1. Display current settings for trigger-related parameters.
 - Clear the display screen.
 - Display the source.
 - Display the slope.
 - Display the external coupling.
 - Display the external attenuation.
 - Display the mainframe trigger level.
 - Display the delay adjust.

For 11802 only

 - Display the delay line coupling.
 - Display the delay line gain.
2. Prompt the user for which item to change.
3. Take one of the following actions based on the users choice from the keypad.

SOURCE

For 11801 only

- Display selectors for external, internal clock (TDR), freerun, mainframe, or head.
- Prompt the user for the source (for **Tgmode Register** <23> bits 5, 6, 9, and 15).
- If the response is a head then do the following:
 - Prompt the user for the MCU identification.
 - Prompt the user for the channel.
 - Ask the user if the trigger is from a trigger head or strobe sense feedback.

Timebase

For 11802 only

- Display selectors for external, internal clock (TDR), freerun, mainframe, head, delay line 1, or delay line 2.
- Prompt the user for the source (for **Tgmode Register <23>** bits 5, 6, 9, and 15).
- If the response is a head then do the following:
 - Prompt the user for the channel.
 - Ask the user if the trigger is from a trigger head or strobe sense feedback.

SLOPE

- Prompt the user for positive or negative slope (for **Tgmode Register <23>** bit 7).

EXTERNAL COUPLING

- Prompt the user for AC or DC coupling (for **Tgmode Register <23>** bit 12).

EXTERNAL ATTENUATION

- Prompt the user for x1 or x10 attenuation (for **Tgmode Register <23>** bit 8).

MAINFRAME TRIGGER LEVEL

- Prompt the user for the trigger level value (for **Trigger Level DAC <22>**).

TRIGGER DELAY ADJUST

- Prompt the user for the amount of trigger delay adjust (for **Trigger Delay Adjust <23>**).

DELAY LINE COUPLING

- Prompt the user for AC or DC coupling (for **Tgmode Register <23>** bit 10).

DELAY LINE GAIN

- Prompt the user for x1 or x10 gain (for **Tgmode Register <23>** bit 11).

4. Repeat the previous steps until the user exits the menu.

Error Index

None.

Timebase

Strobe Gen

Exerciser

Trigger ()

See Also

The Strobe Gen Exerciser Execute test.

Timebase

Routine Name VCO

Overview This routine prompts the user for VCO-related information. This information is loaded into various VCO-related hardware registers when the exerciser is executed.

Description

1. Display current settings for VCO-related parameters.
 - Clear the display screen.
 - Display coarse frequency DAC.
 - Display fine frequency DAC.
 - Display temperature compensation DAC.
 - Display temperature compensation state.
 - Display VCO power state.
2. Prompt the user for which item to change.
3. Take one of the following actions based on the users choice from the keypad.

COARSE FREQUENCY DAC

- Prompt the user for the new value (for VCO Coarse Frequency Control <23> DAC part b).

FINE FREQUENCY DAC

- Prompt the user for the new value (for VCO Fine Frequency Control <23>).

TEMPERATURE COMPENSATION DAC

- Prompt the user for the new value (for VCO Coarse Frequency Control <23> DAC part a).

TEMPERATURE COMPENSATION STATE

- Prompt the user to set temperature compensation on or off (for VCO Coarse Frequency Control <23> TGMODE4).

VCO POWER STATE

- Prompt the user to set VCO power on or off (for VCO Power Control <23> PSTAT7).

4. Repeat the previous steps until the user exits the menu.

Strobe Gen

Exerciser

VCO ()

Error Index

None.

See Also

The Strobe Gen Exerciser Execute test.

Timebase

Routine Name Int Clk (Internal Clock)

Overview This routine prompts the user for internal clock-related information. This information is loaded into various internal clock-related hardware registers when the exerciser is executed.

Description

1. Display current settings for internal clock-related parameters.
 - Clear the display screen.
 - Display internal clock (TDR) rate.
 - Display TDR drive.
 - Display TDR On/Off selector.
2. Prompt the user for which item to change.
3. Take one of the following actions based on the users choice from the keypad.

INTERNAL CLOCK (TDR) RATE

- Prompt the user for the new value (for **Internal Clock Generator** <16B>).

TDR DRIVE

- Prompt the user for the mode to be used for the TDR drive signal (for **Internal Trigger/TDR Control Register** <16B>).

TDR ON/OFF

For 11801 only

- Prompt the user for the MCU identification.
- Prompt the user for the channel.
- Prompt the user to set TDR on or off (for **Internal Trigger/TDR Control Register** <16B> TDRON).
- Send a TDR On/Off command to the acquisition system that contains the selected channel.

For 11802 only

- Prompt the user for the channel.
- Prompt the user to set TDR on or off (for **Internal Trigger/TDR Control Register** <16B> TDRON).

Timebase

Strobe Gen

Exerciser

Int Clk ()

- Send a TDR On/Off command to the acquisition system that contains the selected channel.

4. Repeat the previous steps until the user exits the menu.

Error Index

None.

See Also

The Strobe Gen Exerciser Execute test.

Timebase

Routine Name	BH Adjust (Between Head Adjust)
Overview	This routine prompts the user for Between Head Delay Adjust related information. This information is loaded into the selected Between Head Delay Adjust register.
Description	<ol style="list-style-type: none">1. Clear the display screen.2. Display Between Head Adjust selector. For 11801 only<ul style="list-style-type: none">• Prompt the user for the MCU identification.• Prompt the user for the channel.• Prompt the user for the between head adjust value.• Write the new value to the address that corresponds to the channel the user entered (for Between Head Delay Control <16A>). For 11802 only<ul style="list-style-type: none">• Prompt the user for the channel.• Prompt the user for the between head adjust value.• Write the new value to the address that corresponds to the channel the user entered (for Between Head Delay Control <16A>).3. Repeat the previous steps until the user exits the menu.
Error Index	None.
See Also	The Strobe Gen Exerciser Execute test.

Routine Name Execute

Overview This routine uses the various pieces of information collected in the other routines in this area and loads that data into the associated registers. Next, this routine attempts to initiate the sampling process or self oscillation.

Description

- Place the user-specified values into the VCO registers except for the temperature compensation state, which will be loaded when the **Tgmode Register <23>** is loaded (TGMODE4).
 - Write coarse frequency DAC value to **VCO Coarse Frequency Control <23> DAC part b**.
 - Write fine frequency DAC value to **VCO Fine Frequency Control <23>**
 - Write temperature compensation DAC value to **VCO Coarse Frequency Control <23> DAC part a**.
 - If the VCO is to be turned on, then set **VCO Power Control PSTAT7 <23>** high; else set PSTAT7 low.

- Write an inverted copy of the internal clock rate value to **Internal Clock Generator <16B>**.

- From the table below, write an appropriate value to **Internal Trigger/TDR Control Register <16B>**.

Source	PRETRIG	LOOPGAIN ON	TDRON
TDR Drive On	X	1	0
TDR Drive Off	X	0	1
TDR Drive Divide 2	X	1	1
Internal Clock on	0	X	X
Internal Clock off	1	X	X

- Write FF_{hex} to **Trigger Level DAC <22>**.
- Write $60A2_{hex}$ to **Tgmode Register <23>**.
- Write 5383_{hex} to **Amode Register <18>**.
- Write the Trigger Delay Adjust value to **Trigger Delay Adjust U1120 <23>**.

INTERNAL CLOCK TRIGGER, EXTERNAL TRIGGER, TRIGGER HEAD, OR
DELAY LINES

- If a trigger head has been selected, then send the acquisition system containing the selected head a "trigger head on" command.

9. Set up **Programmable Delay Generator (slow)** U1250 <21> with the Initial Dot Delay, Dot Delay, and Holdoff values.
10. Pulse HOF_LEAST(L) to load the values into their associated registers.
11. Initialize **Programmable Delay Generator (fast)** <23> U1450 by setting pins 50 & 20 (BD0, BD1) high and pulsing pin 12 (F_CNT_LD).
12. Set **Amode Register** <18> bit 12 (MODE12) high if leading edge was selected and low if trailing edge was selected.
13. Set **Tgmode Register** <23> bits 5, 6, 7, 8, 9, 10, 11, 12, 15 with values derived from the TGMODE REGISTER TABLES below, based on previous user selections.
14. Write the Trigger Level value to **Trigger Level DAC** <22>.
15. Load **Programmable Timers** <17> Timer 0 Max A Register with the Number of Dummy Strokes, Max B Register with the Number of Strokes, Counter Register with 0, and Mode/Control Word Register with 4000_{hex}.
16. Write C006_{hex} to Timer 0 Mode/Control Word Register.
17. Wait an amount of time equal to the amount of time it should take to complete the programmed acquisition cycle.
18. Check Timer 0 Mode/Control Word Register bit 15. If the bit is high, then display a message indicating that the timebase is triggered; else display a message indicating that the timebase is not triggered.
19. Check if the user wants to terminate the test. If the user does not want to terminate the test, then go to step 9.

FREERUN TRIGGER

8. Write 04_{hex} to M/F Strobe Sense Feed Back Control <16A>.
9. Write FF_{hex} to Trigger Level DAC <22>.
10. Set **Trigger Recognizer** <23> TGMODE0 high.
11. Set up **Programmable Delay Generator (slow)** U1250 <21> with the Initial Dot Delay, Dot Delay, and Holdoff values.
12. Pulse HOF_LEAST(L) to load the values into their associated registers.
13. Initialize **Programmable Delay Generator (fast)** <23> U1450 by setting pins 50 & 20 (BD0, BD1) high and pulsing pin 12 (F_CNT_LD).

Timebase

14. Set **Amode Register** <18> bit 12 (MODE12) high if leading edge was selected and low if trailing edge was selected.
15. Load **Programmable Timers** <17> Timer 0 Max A Register with the Number of Dummy Strokes, Max B Register with the Number of Strokes, Counter Register with 0, and Mode/Control Word Register with 4000_{hex} .
16. Pulse **Variable Holdoff** TG_ARM(L) <22>.
17. Set **Tgmode Register** <23> bits 5, 6, 7, 8, 9, 10, 11, 12, 15 with values derived from the TGMODE REGISTER TABLES below, based on previous user selections.
18. Pulse **Trig Source Selector & Amplifier** TG_PULSE(L) <22> three times.
19. Write $C005_{hex}$ to Timer 0 Mode/Control Word Register and wait an amount of time equal to the amount of time it should take to complete the programmed acquisition cycle.
20. Check Timer 0 Mode/Control Word Register bit 15. If the bit is high, then display a message indicating that the timebase is triggered; else display a message indicating that the timebase is not triggered.
21. Check if the user wants to terminate the test. If the user does not want to terminate the test, then go to step 11.

STROBE SENSE FEEDBACK TRIGGER

8. Write the Trigger Level value to **Trigger Level DAC** <22>.
9. Enable the strobe sense feedback from the selected channel by writing the appropriate value to **M/F Strobe Sense Feed Back Control** <16A>.
10. Send a "trigger head off" command to the acquisition system that contains the selected channel.
11. Set **Trigger Recognizer** <23> TGMODE0 low .
12. Set **Amode Register** <18> MODE12, MODE14, MODE15 to 0, 1, and 0 respectively.
13. Set up **Programmable Delay Generator (slow)** U1250 <21> with the Initial Dot Delay, Dot Delay, and Holdoff values.
14. Pulse HOF_LEAST(L) to load the values into their associated registers.

Timebase

15. Initialize **Programmable Delay Generator (fast)** <23> U1450 by setting pins 50 & 20 (BD0, BD1) high and pulsing pin 12 (F_CNT_LD).
16. Set **Amode Register** <18> bit 12 (MODE12) high if leading edge was selected and low if trailing edge was selected.
17. Load **Programmable Timers** <17> Timer 0 Max A Register with the Number of Dummy Strokes, Max B Register with the Number of Strokes, Counter Register with 0, and Mode/Control Word Register with 4000hex.
18. Set **Tgmode Register** <23> bits 5, 6, 7, 8, 9, 10, 11, 12, 15 with values derived from the TGMODE REGISTER TABLES below, based on previous user selections.
19. Pulse **Variable Holdoff** TG_ARM(L) <22>.
20. Pulse **Trig Source Selector & Amplifier** TG_PULSE(L) <22> four times.
21. Write C005hex to Timer 0 Mode/Control Word Register and wait an amount of time equal to the amount of time it should take to complete the programmed acquisition cycle.
22. Check Timer 0 Mode/Control Word Register bit 15. If the bit is high, then display a message indicating that the timebase is triggered; else display a message indicating that the timebase is not triggered.
23. Check if the user wants to terminate the test. If the user does not want to terminate the test, then go to step 13.

TGMODE REGISTER TABLES

Source	TGMODE5	TGMODE6	TGMODE9	TGMODE15
Delay Line 1	0	0	0	1
Delay Line 2	0	0	1	1
External Input	0	0	X	0
Internal Clock	0	1	X	X
Head Sense	1	X	X	X
Trigger Head	1	X	X	X

Slope	TGMODE7
Positive	1
Negative	0

External Coupling	TGMODE12	External Attenuation	TGMODE8
AC	1	x1	0
DC	0	x10	1

Delay Line Coupling	TGMODE10	Delay Line Gain	TGMODE11
AC	1	x1	0
DC	0	x10	1

Error Index

None.

See Also

The previous tests in this area for pertinent register setup values.

M/F I/F	Comm	Acq [1-2] (T31[1-2]X)
MCU A	Comm	Acq [1-8] (T41[1-8]X)
MCU B	Comm	Acq [1-8] (T51[1-8]X)
MCU C	Comm	Acq [1-8] (T61[1-8]X)
MCU D	Comm	Acq [1-8] (T71[1-8]X)

Routine Name Acq [1-8]

Overview This test verifies the communications path (TBC-ACQ Data Transceivers <20>, TBC-ACQ Address Buffers <20>, and TBC-ACQ Control Logic <20>) between the Timebase processor and the selected Acquisition processor. The Timebase processor sends a known test pattern and subsequent interrupt to the selected Acquisition processor. The Acquisition processor verifies what was received, inverts the data, and then sends it back to the Timebase processor. The Timebase processor then verifies the inverted data.

Description

- Write the test patterns to the communications buffer of the selected Acquisition processor.
 - Write the test patterns 0006hex, 0001hex, 0002hex, 0004hex, 0008hex, 0010hex, 0020hex, 0040hex, 0080hex, 0100hex, 0200hex, 0400hex, 0800hex, 1000hex, 2000hex, 4000hex to the memory locations starting at an offset of 2BA8hex above the *Memory* address given in the appropriate Address Table below.
- Initialize the interrupt acknowledgement memory location.
 - Write 5A5Ahex to the memory location starting at an offset of 2B9Ehex above the *Memory* address given in the appropriate Address Table below.
- Send an interrupt to the selected Acquisition processor.
 - Generate the interrupt by writing a 0000hex to the *Interrupt* address given in the appropriate Address Table below.
- Wait for 10 ms.
- Read the communications buffer (at same locations as step 1) and verify the contents.
 - The buffer should now contain: 0006hex, FFFEhex, FFFDhex, FFFBhex, FFF7hex, FFEFhex, FFDFhex, FFBFhex, FF7Fhex, FEFFhex, FDFFhex, FBFFhex, F7FFhex, EFFFhex, DFFFhex, BFFFhex.
- Verify that the interrupt acknowledgement memory location (written by the Acquisition processor) is correct.
 - The memory location (from step 2) that was previously 5A5Ahex should now contain A5A5hex.
- Verify that the group interrupt status bit is set for the selected Acquisition processor.

Timebase

M/F I/F	Comm	Acq [1-2] (T31[1-2]X)
MCU A	Comm	Acq [1-8] (T41[1-8]X)
MCU B	Comm	Acq [1-8] (T51[1-8]X)
MCU C	Comm	Acq [1-8] (T61[1-8]X)
MCU D	Comm	Acq [1-8] (T71[1-8]X)

- Read the memory location at an offset of 0002_{hex} above the *Interrupt* address given in the appropriate Address Table below and verify that the appropriate *Status* bit is set.
8. Reset the interrupt acknowledgement memory location by performing step 2 again.
 9. Reset the selected Acquisition processor's **TBC Interrupt Latch** <14>.

Error Index
T[3-7]1[1-8]2

The first word of the message returned by the Acquisition processor in the communications buffer (i.e. the command word) was not the expected pattern.

Error Index
T[3-7]1[1-8]3

One or more of the test patterns in the communications buffer, starting at the second word, returned by the Acquisition processor was not the expected pattern.

Error Index
T[3-7]1[1-8]4

The expected interrupt was not received from the selected Acquisition processor.

Error Index
T[3-7]1[1-8]5

The interrupt returned was not from the selected Acquisition processor.

Error Index ????

The selected Acquisition processor is not present or is not communicating with the Timebase processor.

Timebase

M/F	I/F	Comm	Acq [1-2] (T31[1-2]X)
MCU A		Comm	Acq [1-8] (T41[1-8]X)
MCU B		Comm	Acq [1-8] (T51[1-8]X)
MCU C		Comm	Acq [1-8] (T61[1-8]X)
MCU D		Comm	Acq [1-8] (T71[1-8]X)

Address Table for 11801

	M/F	MCU A	MCU B	MCU C	MCU D
Acq 1					
Memory	20000	40000	60000	80000	A0000
Interrupt	30200	30240	30260	30280	302A0
Status	bit 0	bit 0	bit 8	bit 0	bit 8
Acq 2					
Memory	24000	44000	64000	84000	A4000
Interrupt	30204	30244	30264	30284	302A4
Status	bit 1	bit 1	bit 9	bit 1	bit 9
Acq 3					
Memory		48000	68000	88000	A8000
Interrupt		30248	30268	30288	302A8
Status		bit 2	bit 10	bit 2	bit 10
Acq 4					
Memory		4C000	6C000	8C000	AC000
Interrupt		3024C	3026C	3028C	302AC
Status		bit 3	bit 11	bit 3	bit 11
Acq 5					
Memory		50000	70000	90000	B0000
Interrupt		30250	30270	30290	302B0
Status		bit 4	bit 12	bit 4	bit 12
Acq 6					
Memory		54000	74000	94000	B4000
Interrupt		30254	30274	30294	302B4
Status		bit 5	bit 13	bit 5	bit 13
Acq 7					
Memory		58000	78000	98000	B8000
Interrupt		30258	30278	30298	302B8
Status		bit 6	bit 14	bit 6	bit 14
Acq 8					
Memory		5C000	7C000	9C000	BC000
Interrupt		3025C	3027C	3029C	302BC
Status		bit 7	bit 15	bit 7	bit 15

Timebase

M/F I/F	Comm	Acq [1-2] (T31[1-2]X)
MCU A	Comm	Acq [1-8] (T41[1-8]X)
MCU B	Comm	Acq [1-8] (T51[1-8]X)
MCU C	Comm	Acq [1-8] (T61[1-8]X)
MCU D	Comm	Acq [1-8] (T71[1-8]X)

Address Table for 11802

Acq 1	
Memory	24000
Interrupt	30204
Status	bit 1

Timebase

M/F I/F	Shared Mem	Acq [1-2] (T32[1-2]X)
MCU A	Shared Mem	Acq [1-8] (T42[1-8]X)
MCU B	Shared Mem	Acq [1-8] (T52[1-8]X)
MCU C	Shared Mem	Acq [1-8] (T62[1-8]X)
MCU D	Shared Mem	Acq [1-8] (T72[1-8]X)

Routine Name Acq [1-8]

Overview This test verifies the data and address lines of the Acquisition Shared RAM <12> from the Timebase processor. The data lines are verified by performing a "walking zero's" test on one memory location. The address lines are tested by performing a simple RAM test on all 16K bytes of shared memory (the RAM itself is more thoroughly checked by the appropriate Acquisition processor).

Description

1. Perform a "walking zero's" test on **Shared RAM** <12> at the *Memory* location given in the appropriate Address Table below for the selected Acquisition processor. Terminate the test if any verify operation fails.
 - Write the pattern 7FFFhex to the appropriate address. Read the same address and verify that it was 7FFFhex. Continue this write/read/verify sequence with the patterns BFFFhex, DFFFhex, EFFFhex, F7FFhex, FBFFhex, FDFFFhex, FEFFhex, FF7Fhex, FFBFhex, FFDFhex, FFEFhex, FFF7hex, FFFBhex, FFFDhex, FFFEhex.
2. Verify the addressability of the lower 10K bytes of **Shared RAM** <12> by performing a RAM test on the specified memory locations (use the appropriate *Memory* location in the Address Table as the base memory location to calculate the memory locations). Terminate the test if any verify operation fails.
 - Fill the address range *Memory* to *Memory* + 27FFhex with the pattern AAAAhex.
 - Read and verify that the first *Memory* location contains AAAAhex. If so, write 5555hex to the first *Memory* location. Increment the address and continue this read/verify/write sequence until address *Memory* + 27FFhex is reached.
 - Repeat the previous read/verify/write sequence, starting again at the first *Memory* location, with patterns 5555hex and 0000hex (i.e., reading and verifying the previous pattern written and then writing the next pattern).
3. Save the upper 6K bytes of **Shared RAM** <12> into the lower portion of **Shared RAM** by copying the data from address range *Memory* + 2800hex to *Memory* + 3FFFhex into the locations starting at address *Memory*.
4. Verify the addressability of the upper 6K bytes of **Shared RAM** <12> by performing the same procedures as in step 2 for address range *Memory* + 2800hex to *Memory* + 3FFFhex.
5. Restore the upper 6K bytes of **Shared RAM** <12> by copying the lower 6K bytes of **Shared RAM** into the upper 6K bytes.

Timebase

M/F I/F	Shared Mem	Acq [1-2] (T32[1-2]X)
MCU A	Shared Mem	Acq [1-8] (T42[1-8]X)
MCU B	Shared Mem	Acq [1-8] (T52[1-8]X)
MCU C	Shared Mem	Acq [1-8] (T62[1-8]X)
MCU D	Shared Mem	Acq [1-8] (T72[1-8]X)

Error Index
T[3-7]2[1-8]1

One of the "walking zero's" patterns read from the **Shared RAM** <12> did not match the value written while performing the data line test.

Error Index
T[3-7]2[1-8]2

One of the patterns read from the **Shared RAM** <12> did not match the pattern written while performing the address line test.

Error Index ????

The selected Acquisition processor is not present or is not communicating with the Timebase processor.

Address Table for 11801

	M/F	MCU A	MCU B	MCU C	MCU D
Acq 1 Memory	20000	40000	60000	80000	A0000
Acq 2 Memory	24000	44000	64000	84000	A4000
Acq 3 Memory		48000	68000	88000	A8000
Acq 4 Memory		4C000	6C000	8C000	AC000
Acq 5 Memory		50000	70000	90000	B0000
Acq 6 Memory		54000	74000	94000	B4000
Acq 7 Memory		58000	78000	98000	B8000
Acq 8 Memory		5C000	7C000	9C000	BC000

Address Table for 11802

	M/F
Acq 1 Memory	24000

Timebase

M/F I/F	Strobe Gate	Acq [1-2] (T33[1-2]X)
MCU A	Strobe Gate	Acq [1-8] (T43[1-8]X)
MCU B	Strobe Gate	Acq [1-8] (T53[1-8]X)
MCU C	Strobe Gate	Acq [1-8] (T63[1-8]X)
MCU D	Strobe Gate	Acq [1-8] (T73[1-8]X)

Routine Name Acq [1-8]

Overview This test verifies **Gated Strobe Control Logic** <20> by simulating the generation of strobe signals.

Description

1. Write 0003*hex* to **Amode Register** <18> to place the timebase into parallel diagnostics mode, thus allowing processor control of **Gated Strobe Control Logic** PCLK <20> (via **PDG Control Logic** <21>). This also sets **Gated Strobe Control Logic** MODE10 <20> low, which inhibits Acquisition systems from digitizing data.
2. Send the selected Acquisition processor an "assign A/D" command to enable its A/D's. If the Acquisition processor does not respond to this command, then terminate the test.
3. Initialize the selected Acquisition processor's **Shared RAM** <12>.
 - Write 5AA5*hex* to memory locations with offsets of 0002*hex*, 0004*hex*, 0006*hex*, 0008*hex*, and 0010*hex* above the base *Memory* location given in the appropriate Address Table below.
4. Set up **Programmable Timers** <17> Timer 0.
 - Write 0002*hex* to **Tgmode Register** <23>.
 - Write 0000*hex* to Counter Register, 0001*hex* to Max A Register, 0004*hex* to Max B Register, and C007*hex* to Mode/Control Word Register.
5. Generate a dummy strobe to allow **Gated Strobe Control Logic** BACQGAT(L) <20> to go low and enable acquisition strobes.
 - Wait 1 ms.
 - Set **Gated Strobe Control Logic** MODE10 <20> high.
 - Pulse **Gated Strobe Control Logic** PCLK <20>.
6. Verify that the selected Acquisition processor's **Shared RAM** <12> has not changed.
 - Read the acquisition memory location at an offset of 0002*hex* above the base *Memory* location given in the appropriate Address Table below and verify that it is still 5AA5*hex*.
7. Generate a real strobe.
 - Pulse **Gated Strobe Control Logic** PCLK <20>.

Timebase

M/F I/F	Strobe Gate	Acq [1-2] (T33[1-2]X)
MCU A	Strobe Gate	Acq [1-8] (T43[1-8]X)
MCU B	Strobe Gate	Acq [1-8] (T53[1-8]X)
MCU C	Strobe Gate	Acq [1-8] (T63[1-8]X)
MCU D	Strobe Gate	Acq [1-8] (T73[1-8]X)

8. Verify that the selected Acquisition processor's Shared RAM <12> has changed and is within expected digitized range.
 - Read the acquisition memory location at an offset of 0002hex above the base Memory location given in the appropriate Address Table below and verify that the low and high bytes are between 60hex and A0hex.
9. Verify that the next memory word in Shared RAM <12> has not changed (i.e. that the last strobe only caused the acquisition to digitize one value into the RAM).
 - Read the acquisition memory location at an offset of 0004hex (i.e. 0002hex + 0002hex) above the base Memory location given in the appropriate Address Table below and verify that it is still 5AA5hex.
10. Repeat steps 7 - 9 for offsets 0004hex, 0006hex, and 0008hex above the base address given in the appropriate Address Table below.
11. Send the selected Acquisition processor an "inhibit acquisition" command.

Error Index T[3-7]3[1-8]1

The selected Acquisition processor is not responding to commands from the Timebase processor as it should.

Error Index T[3-7]3[1-8]2

One of the values in Shared RAM <12> changed (either after the dummy strobe or one of the real strobe pulses) when it shouldn't have. The correspondence of the "Min" field displayed on the screen to the actual memory location which shouldn't have changed (but did) is shown in the following table (using the appropriate Memory location in the Address Table further below as the base memory location to calculate the memory locations).

"Min" Field	Memory Location	After Strobe
0000	Memory + 0002hex	(dummy strobe)
0000	Memory + 0004hex	(real strobe #1)
0001	Memory + 0006hex	(real strobe #2)
0002	Memory + 0008hex	(real strobe #3)
0003	Memory + 0010hex	(real strobe #4)

Error Index T[3-7]3[1-8]3

The digitized value read back from the low byte of Shared RAM <12> was not within the expected range.

Timebase

M/F I/F	Strobe Gate	Acq [1-2] (T33[1-2]X)
MCU A	Strobe Gate	Acq [1-8] (T43[1-8]X)
MCU B	Strobe Gate	Acq [1-8] (T53[1-8]X)
MCU C	Strobe Gate	Acq [1-8] (T63[1-8]X)
MCU D	Strobe Gate	Acq [1-8] (T73[1-8]X)

Error Index
T[3-7]3[1-8]4

The digitized value read back from the high byte of Shared RAM <12> was not within the expected range.

Error Index ????

The selected Acquisition processor is not present or is not communicating with the Timebase processor.

Address Table for 11801

	M/F	MCU A	MCU B	MCU C	MCU D
Acq 1 Memory	20000	40000	60000	80000	A0000
Acq 2 Memory	24000	44000	64000	84000	A4000
Acq 3 Memory		48000	68000	88000	A8000
Acq 4 Memory		4C000	6C000	8C000	AC000
Acq 5 Memory		50000	70000	90000	B0000
Acq 6 Memory		54000	74000	94000	B4000
Acq 7 Memory		58000	78000	98000	B8000
Acq 8 Memory		5C000	7C000	9C000	BC000

Address Table for 11802

	M/F
Acq 1 Memory	24000

Timebase

M/F I/F	Strobe Sense	Acq [1-2] (T34[1-2]X)
MCU A	Strobe Sense	Acq [1-8] (T44[1-8]X)
MCU B	Strobe Sense	Acq [1-8] (T54[1-8]X)
MCU C	Strobe Sense	Acq [1-8] (T64[1-8]X)
MCU D	Strobe Sense	Acq [1-8] (T74[1-8]X)

Routine Name Acq [1-8]

Overview This test verifies **Between Head Delay Adjust** <15> (<7> for SM-11 Multi-Channel Units), **Between Head Delay Control** <16A> (<6> for SM-11's), and **Strobe Sense Feedback Control** <16A> (<5> for SM-11's) by taking loop delay time samples every half volt step of the appropriate control signal over the entire range of the selected delay device and then applying a least-squares fit to the data.

Description

1. Check the selected Acquisition system's **Shared RAM** <12> to see if one or more sampling heads are plugged into the acquisition system. If no heads are plugged in, then abort the test.
2. Set the **VCO** <23> to its default values.
 - Set **VCO Coarse Frequency Control** <23> DAC part a to 80hex and DAC part b to B4hex.
 - Set **VCO Fine Frequency Control** <23> to 800hex.
 - Set **Delay Adjust DAC** <23> to 800hex.
3. Set the delay control of **Between Head Delay Adjust** <15> (<7> for SM-11's).
 - Write the first test pattern from the following table to the **Between Head Delay Control** <16A> (<6> for SM-11's) course and fine DACs (only the course DAC is exercised) for the part of the DACs which correspond to the first sampling head found to be present in step 1.

Test Pattern (hex)

0000
1000
2000
3000
4000
5000
6000
7000
8000
9000
A000
B000
C000
D000
E000
F000

Timebase

M/F I/F	Strobe Sense	Acq [1-2] (T34[1-2]X)
MCU A	Strobe Sense	Acq [1-8] (T44[1-8]X)
MCU B	Strobe Sense	Acq [1-8] (T54[1-8]X)
MCU C	Strobe Sense	Acq [1-8] (T64[1-8]X)
MCU D	Strobe Sense	Acq [1-8] (T74[1-8]X)

4. Measure the VCO frequency.

- Write `6020hex` to **Tgmode Register <23>**. This sets VCO U1330-4 TGMODE4 low to connect the VCO Coarse Frequency Control DAC outputs to the VCO Vcon input. It also sets up **Programmable Delay Generator (fast) U1450 GATED COUNTER <23>** to use FRQGATE(L) as a gate for clock input CLK1 on pins 67 & 68 (from VCO U1430 COUT/COUTB). This allows **Slow Frequency Counter VCLK <23>** to drive **Programmable Timers U440-20 TMRIN0 <17>** via **Reference Clock & Frequency/Period Measurement Logic U160 <18>**.
- Write `0003hex` to **Amode Register <18>** to set PDG Control Logic and **Programmable Delay Generator (slow) <21>** into parallel diagnostics mode.
- Load **Programmable Timers <17>** Timer 1 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with `4000hex`.
- Load **Programmable Timers <17>** Timer 0 Max A Register with `FFFFhex`, Counter Register with 0, and Mode/Control Word Register with `4000hex`.
- Pulse RESCON(L) to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, **Programmable Delay Generator (fast) <23>**, and **Slow Frequency Counter <23>**.
- Write `C005hex` to Timer 0 Mode/Control Word Register.
- Wait 1 ms for the VCO to stabilize.
- Write `C006hex` to Timer 1 Mode/Control Word Register to start the VCO frequency measurement operation. This causes **Reference Clock & Frequency/Period Measurement Logic U360B-12 BTMROUT1** and **FRQGATE(L) <18>** to eventually go low, allowing the clock input of **Programmable Delay Generator (fast) GATED COUNTER <23>** to drive **Slow Frequency Counter VCLK <23>**, which is essentially counted via **Programmable Timers TMRIN0 <17>**.
- Wait 2 ms to allow the frequency measurement to stop via **FRQGATE(L)** going back high.
- Verify that **Programmable Timers Timer 1 <17>** terminated properly in order to make a valid VCO frequency measurement. If so, retain the count in Timer 0.

Timebase

M/F I/F	Strobe Sense	Acq [1-2] (T34[1-2]X)
MCU A	Strobe Sense	Acq [1-8] (T44[1-8]X)
MCU B	Strobe Sense	Acq [1-8] (T54[1-8]X)
MCU C	Strobe Sense	Acq [1-8] (T64[1-8]X)
MCU D	Strobe Sense	Acq [1-8] (T74[1-8]X)

- Repeat the previous items of this step seven more times, average the results, and save this VCO frequency value for the following loop delay time calculations.
5. Measure the loop delay time with **Programmable Delay Generator Generator (fast) STRP <23>** (i.e. the holdoff delay) as the delay feedback element.
- Set **Trigger Level DAC U800 <22>** to 4Chex (0.4 volts).
 - Enable the strobe sense from the sampling head of the selected Acquisition system by writing the *Head Select* value from the appropriate Address Table below to the **Acquisition Interface Logic latch <16A> (<4> for SM-11's)** which controls the **Strobe Sense Feedback Control** sense enable lines.
 - Write 60A2hex to **Tgmode Register <23>** and 5387hex to **Amode Register <18>** to stop strobes and set up timebase.
 - Load **Programmable Timers <17>** Timer 0 Max A Register with 10, Max B Register with 1024, Counter Register with 0, and Mode/Control Word Register with 4000hex .
 - Load **Programmable Timers <17>** Timer 1 Max A Register with FFFFhex , Counter Register with 0, and Mode/Control Word Register with 4000hex .
 - Pulse **RESCON(L)** to reset **Reference Clock & Frequency/Period Measurement Logic <18>**, **Programmable Delay Generator (fast) <23>**, and **Slow Frequency Counter <23>**.
 - For each **Programmable Delay Generator (slow)** register given in the following table, set **PDG Control Logic MODE8** and **MODE9** to the given levels and use the specified clock to transfer the given initial pattern into the register. This is done by shifting each bit, via **PDG Control Logic BD0**, into the register by pulsing the appropriate clock signal 24 or 48 times (depending on the length of the register).

U1250 Shadow Register	Pattern (hex)	Clock	MODE9	MODE8
STROBE DELAY	FFFFFFA9000000	BCLK	0	1
HOLD OFF DELAY	FFFFFFE	HHCLK	1	0
DOT DELAY	00000000000000	DCLK	0	0

- Pulse **Programmable Delay Generator (slow) HOF_LEAST(H)** to transfer the internal shadow registers of U1250 into the next register stages. This also sets the least 6 bits of the overall 30-bit

Timebase

M/F I/F
MCU A
MCU B
MCU C
MCU D

Strobe Sense
Strobe Sense
Strobe Sense
Strobe Sense
Strobe Sense

Acq [1-2] (T34[1-2]X)
Acq [1-8] (T44[1-8]X)
Acq [1-8] (T54[1-8]X)
Acq [1-8] (T64[1-8]X)
Acq [1-8] (T74[1-8]X)

holdoff value to an initial value of 29_{hex} by loading $A4_{hex}$ into Holdoff Register U941 <23>.

- **Pulse Programmable Delay Generator (slow)** HLD and YLD to transfer the HOLDOFF DELAY register into the HOLDOFF COUNTER and the STROBE DELAY register into the STROBE DELAY COUNTER.
 - **Enable Programmable Delay Generator (fast)** U1450 holdoff delay and strobe delay load signals by setting BD0 and BD1 high and pulsing F_CNT_LD.
 - Write 4386_{hex} to Amode Register <18> to select the holdoff as the delay feedback element (via Variable Holdoff STRP <22>) and to set the timebase into normal mode of operation (mode bits 0 & 1).
 - Write $60A2_{hex}$ to Tgmode Register <23> to inhibit freerun mode in the Trigger Recognizer <22>.
 - **Pulse Variable Holdoff** TG_ARM(L) <22> and then **Trig Source Selector & Amplifier** TG_PULSE(L) <22> to allow triggering to occur correctly.
 - Write $C005_{hex}$ to Timer 1 Mode/Control Word Register and $C006_{hex}$ to Timer 0 Mode/Control Word Register to start the delay measurement operation.
 - Wait approximately 3 ms.
 - Verify that **Programmable Timers** Timer 0 <17> terminated properly in order to make a valid delay measurement. If so, use the previously saved VCO frequency measurement in step 4 to calculate the actual loop delay time. Retain this calculated loop delay value.
 - Repeat the previous items of this step seven more times, average the results, and save for use in step 7.
6. Repeat steps 3, 4, and 5 for the remaining test patterns in the table of step 3.
 7. Perform a least-squares fit of the collected loop delay times and verify that the calculated least-squares error and slope are within expected ranges.
 8. If two sampling heads are present for the selected Acquisition system, then repeat the test (steps 2-7) for the other sampling head.

Timebase

M/F I/F	Strobe Sense	Acq [1-2] (T34[1-2]X)
MCU A	Strobe Sense	Acq [1-8] (T44[1-8]X)
MCU B	Strobe Sense	Acq [1-8] (T54[1-8]X)
MCU C	Strobe Sense	Acq [1-8] (T64[1-8]X)
MCU D	Strobe Sense	Acq [1-8] (T74[1-8]X)

Error Index
T[3-7]4[1-8]1

The calculated least-squares error of the loop delay times for sampling head 1 of the selected Acquisition system was not with the expected range.

Error Index
T[3-7]4[1-8]2

The calculated least-squares error of the loop delay times for sampling head 2 of the selected Acquisition system was not with the expected range.

Error Index
T[3-7]4[1-8]3

The calculated least-squares slope of the loop delay times for sampling head 1 of the selected Acquisition system was not with the expected range.

Error Index
T[3-7]4[1-8]4

The calculated least-squares slope of the loop delay times for sampling head 2 of the selected Acquisition system was not with the expected range.

Error Index
T[3-7]4[1-8]5

The VCO frequency could not be measured.

Error Index
T[3-7]4[1-8]6

The loop delay time could not be measured.

Error Index ????

The selected Acquisition processor is not present or is not communicating with the Timebase processor; or there are no sampling heads plugged into the selected Acquisition system.

Timebase

M/F I/F
MCU A
MCU B
MCU C
MCU D

Strobe Sense
Strobe Sense
Strobe Sense
Strobe Sense
Strobe Sense

Acq [1-2] (T34[1-2]X)
Acq [1-8] (T44[1-8]X)
Acq [1-8] (T54[1-8]X)
Acq [1-8] (T64[1-8]X)
Acq [1-8] (T74[1-8]X)

Address Table for 11801

		M/F	MCU A	MCU B	MCU C	MCU D
Acq 1	Head 1 Select	0	10	20	30	40
	Head 1 Delay Adjust	2F000	2F100	2F200	2F300	2F400
	Head 2 Select	1	11	21	31	41
	Head 2 Delay Adjust	2F010	2F110	2F210	2F310	2F410
Acq 2	Head 1 Select	2	12	22	32	42
	Head 1 Delay Adjust	2F020	2F120	2F220	2F320	2F420
	Head 2 Select	3	13	23	33	43
	Head 2 Delay Adjust	2F030	2F130	2F230	2F330	2F430
Acq 3	Head 1 Select		14	24	34	44
	Head 1 Delay Adjust		2F140	2F240	2F340	2F440
	Head 2 Select		15	25	35	45
	Head 2 Delay Adjust		2F150	2F250	2F350	2F450
Acq 4	Head 1 Select		16	26	36	46
	Head 1 Delay Adjust		2F160	2F260	2F360	2F460
	Head 2 Select		17	27	37	47
	Head 2 Delay Adjust		2F170	2F270	2F370	2F470
Acq 5	Head 1 Select		18	28	38	48
	Head 1 Delay Adjust		2F180	2F280	2F380	2F480
	Head 2 Select		19	29	39	49
	Head 2 Delay Adjust		2F190	2F290	2F390	2F490
Acq 6	Head 1 Select		1A	2A	3A	4A
	Head 1 Delay Adjust		2F1A0	2F2A0	2F3A0	2F4A0
	Head 2 Select		1B	2B	3B	4B
	Head 2 Delay Adjust		2F1B0	2F2B0	2F3B0	2F4B0
Acq 7	Head 1 Select		1C	2C	3C	4C
	Head 1 Delay Adjust		2F1C0	2F2C0	2F3C0	2F4C0
	Head 2 Select		1D	2D	3D	4D
	Head 2 Delay Adjust		2F1D0	2F2D0	2F3D0	2F4D0
Acq 8	Head 1 Select		1E	2E	3E	4E
	Head 1 Delay Adjust		2F1E0	2F2E0	2F3E0	2F4E0
	Head 2 Select		1F	2F	3F	4F
	Head 2 Delay Adjust		2F1F0	2F2F0	2F3F0	2F4F0

Timebase

M/F I/F	Strobe Sense	Acq [1-2] (T34[1-2]X)
MCU A	Strobe Sense	Acq [1-8] (T44[1-8]X)
MCU B	Strobe Sense	Acq [1-8] (T54[1-8]X)
MCU C	Strobe Sense	Acq [1-8] (T64[1-8]X)
MCU D	Strobe Sense	Acq [1-8] (T74[1-8]X)

Address Table for 11802

Acq 1	Head 1 Select	2
	Head 1 Delay Adjust	2F020
	Head 2 Select	3
	Head 2 Delay Adjust	2F030

Timebase

M/F I/F	Arbitor Test	Acq [1-2] ()
MCU A	Arbitor Test	Acq [1-8] ()
MCU B	Arbitor Test	Acq [1-8] ()
MCU C	Arbitor Test	Acq [1-8] ()
MCU D	Arbitor Test	Acq [1-8] ()

Routine Name Acq [1-8]

Overview This test provides stimulus for debugging the acquisition interface by performing a series of "walking one's" tests on memory locations in Shared RAM <12>. The user can select byte, even word, or odd word accesses to the memory.

The arbitor test, in order to troubleshoot the Timebase-to-Acquisition interface, is meant to be used with a special Acquisition power-up arbitor test loop which is induced by the placement of two jumpers on the Acquisition MPU board.

The description section below describes both the Timebase Arbitor Test and the special Acquisition power-up arbitor test loop.

Operator Procedure The Timebase Arbitor Test requires operator interaction and may only be executed in the "Routine" menu with the "All" and "Loop" selector modes set to "Off". Once this test is invoked, the operator is prompted for relevant test information.

For the failed Acquisition unit:

- Power down the instrument and extend the failed Acquisition unit MPU board by using the Acquisition System and Board Extenders (or the Acquisition Board Extender and extender cables for SM-11's, if possible). If the fault happens to lie in the Acquisition unit of an SM-11, it is possible to put that Acquisition in an extended mainframe Acquisition cage for troubleshooting.
- Place a jumper on J210 of the Acquisition MPU board. This causes the Acquisition processor to enter a special arbitor test loop on power-up instead of executing the Timebase communication test (see Acquisition kernel test description 3).
- Power up the instrument and then place a jumper on J110. This enables Acquisition-to-Timebase interrupt generation, which must be done only after power-on reset, so that it can be checked.

Description TIMEBASE

1. Ask the user for testing modes.
 - Ask the user for 1 of 3 access modes: byte, even word, or odd word.
 - Ask the user for verbose or terse display mode.
2. Write a zero to the *Interrupt* location given in the appropriate Address Table below for the selected Acquisition system.

Timebase

M/F I/F	Arbitor Test	Acq [1-2] ()
MCU A	Arbitor Test	Acq [1-8] ()
MCU B	Arbitor Test	Acq [1-8] ()
MCU C	Arbitor Test	Acq [1-8] ()
MCU D	Arbitor Test	Acq [1-8] ()

- Perform one of the following series of "walking one's" tests, based on the access mode (use the appropriate *Memory* location in the Address Table below as the base memory location to calculate the memory locations). Log any faults and display them if verbose mode is selected. (Test patterns are 01hex, 02hex, ..., 80hex for byte accesses and 0001hex, 0002hex, ..., 8000hex for word accesses.)

Byte Access	Odd Word Access	Even Word Access
<i>Memory</i> + 1400hex	<i>Memory</i> + 1401hex	<i>Memory</i> + 1400hex
<i>Memory</i> + 1401hex	<i>Memory</i> + 1403hex	<i>Memory</i> + 1402hex
<i>Memory</i> + 1402hex	<i>Memory</i> + 1405hex	<i>Memory</i> + 1404hex
<i>Memory</i> + 1403hex	<i>Memory</i> + 1407hex	<i>Memory</i> + 1406hex
•	•	•
•	•	•
•	•	•
<i>Memory</i> + 27FFhex	<i>Memory</i> + 27FDhex	<i>Memory</i> + 27FEhex

- Display the summary of the pass though memory.
- Repeat steps 2, 3, and 4 until the user exits the test.

NOTE

Asynchronous interrupts from Acquisition units which failed the power-up communication test with the Timebase processor are cleared as they occur. This is done by reading the *Interrupt* + 0002hex memory location given in the appropriate Address Table below.

ACQUISITION

- Enable the A/D converters for both channels.
 - Enable Chan A/D A and B <9> by setting **Measurement & A/D Strobe Generator** U901 pins 16 and 17 <13> high.
- Enable local strobing.
 - Enable the processor-generated strobe (MPUSTB) by setting **Enable Register** U801-18 MSTREN(L) <13> low.
- Turn all Acquisition MPU board LEDs off.
 - Write FFhex to **Front Panel LEDs/Switches** U100 <12>.

Timebase

M/F I/F	Arbitor Test	Acq [1-2] ()
MCU A	Arbitor Test	Acq [1-8] ()
MCU B	Arbitor Test	Acq [1-8] ()
MCU C	Arbitor Test	Acq [1-8] ()
MCU D	Arbitor Test	Acq [1-8] ()

4. Generate an interrupt to the Timebase processor if **Front Panel LEDs/Switches J110 <12>** is connected (i.e. shorted).
 - Write *01hex* to address *6C10hex*. This sets **TBC Interrupt Latch U210A-5 <14>** high.
5. If an interrupt is present from the Timebase processor, then clear the Timebase interrupt.
 - If **Interrupt Latch U1111-7 <13>** is high (i.e. a Timebase interrupt is present), then toggle **Interrupt Latch U1111-5 CLRIRQ <13>** by writing to address *6C00hex*. This sets U1111-7 low and clears the interrupt.
6. Turn top LED of Acquisition MPU board "on" to indicate the start of RAM access operations.
 - Set **Panel LEDs/Switches U100-2 <12>** low to turn on DS100.
7. Read all of **Shared RAM <12>** and **Scratchpad RAM <12>** in order to generate address and bus cycles for the **Bus Arbiter <13>** to resolve.
8. Perform a RAM test on memory locations *2C48hex* through *5F9Bhex*.
 - For each memory location, write the lower 8 bits of its address into the memory location, e.g. *48hex* into *2C48hex*, *49hex* into *2C49hex*, etc.
 - Read each memory location, starting at *2C48hex*, and verify that it contains the correct data. If not, then set **Panel LEDs/Switches U100-5 <12>** low to turn on DS101 (i.e. second LED from top of Acquisition MPU board).
9. Turn top LED of Acquisition MPU board "off" to indicate the finish of RAM access operations.
 - Set **Panel LEDs/Switches U100-2 <12>** high to turn off DS100.
10. Clear the waveform address counter.
 - Pulse **Hardware Measurement System U820-13 WACCLR(L) <13>**.
11. Generate 2560 local strobes to acquire waveform data.
 - Pulse **Interrupt Latch U1100B-5 MPUSTB <13>** 2560 times.
12. Repeat steps 4 through 11 forever.

Error Index

None.

Timebase

M/F I/F	Arbitor Test	Acq [1-2] ()
MCU A	Arbitor Test	Acq [1-8] ()
MCU B	Arbitor Test	Acq [1-8] ()
MCU C	Arbitor Test	Acq [1-8] ()
MCU D	Arbitor Test	Acq [1-8] ()

Address Table for 11801

	M/F	MCU A	MCU B	MCU C	MCU D
Acq 1					
Memory	20000	40000	60000	80000	A0000
Interrupt	200	240	260	280	2A0
Acq 2					
Memory	24000	44000	64000	84000	A4000
Interrupt	204	244	264	284	2A4
Acq 3					
Memory		48000	68000	88000	A8000
Interrupt		248	268	288	2A8
Acq 4					
Memory		4C000	6C000	8C000	AC000
Interrupt		24C	26C	28C	2AC
Acq 5					
Memory		50000	70000	90000	B0000
Interrupt		250	270	290	2B0
Acq 6					
Memory		54000	74000	94000	B4000
Interrupt		254	274	294	2B4
Acq 7					
Memory		58000	78000	98000	B8000
Interrupt		258	278	298	2B8
Acq 8					
Memory		5C000	7C000	9C000	BC000
Interrupt		25C	27C	29C	2BC

Address Table for 11802

	M/F
Acq 1	
Memory	24000
Interrupt	204

Timebase

Acq[1-2]	Memory	ROM Loc (m[1-2]11X)
Acq[1-8]	Memory	ROM Loc (a[1-8]11X)
Acq[1-8]	Memory	ROM Loc (b[1-8]11X)
Acq[1-8]	Memory	ROM Loc (c[1-8]11X)
Acq[1-8]	Memory	ROM Loc (d[1-8]11X)

Routine Name ROM Loc (ROM Location)

Overview This test verifies that EPROM U611 <12> is in the correct socket.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations F004_{hex} and F005_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then verify the location byte of the EPROM device.
 - Read and verify the location byte against the known value.

Error Index
[m | a-d][1-8]111 The bytes at F004_{hex} and F005_{hex} were not complementary.

Error Index
[m | a-d][1-8]112 The location byte did not match the known value. The middle four digits of the part number of the faulty EPROM device are displayed in the first four digits of the address field of the results.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Memory	ROM Check (m[1-2]12X)
Acq[1-8]	Memory	ROM Check (a[1-8]12X)
Acq[1-8]	Memory	ROM Check (b[1-8]12X)
Acq[1-8]	Memory	ROM Check (c[1-8]12X)
Acq[1-8]	Memory	ROM Check (d[1-8]12X)

Routine Name ROM Check (ROM Checksum)

Overview This test verifies the integrity of EPROM U611 <12> by performing a checksum on its contents.

Description

1. Perform a complement test to verify that the EPROM is present and can drive the data bus high and low on all bits.
 - Exclusive-or the contents of byte locations F004_{hex} and F005_{hex} and verify that the result is FF_{hex}.
2. If the complement test was successful, then perform a checksum on the contents of the EPROM device.
 - Perform a checksum on all the bytes in U611 except the first two, and then verify it against the checksum stored in the first two bytes of the device.

Error Index
[m | a-d][1-8]121

The bytes at F004_{hex} and F005_{hex} were not complementary.

Error Index
[m | a-d][1-8]122

The computed checksum did not match the stored checksum.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Memory	RAM Data (m[1-2]13X)
Acq[1-8]	Memory	RAM Data (a[1-8]13X)
Acq[1-8]	Memory	RAM Data (b[1-8]13X)
Acq[1-8]	Memory	RAM Data (c[1-8]13X)
Acq[1-8]	Memory	RAM Data (d[1-8]13X)

Routine Name RAM Data

Overview This test verifies the data lines from MPU U401 <12>, through MPU & Bus Buffers <12> and Shared RAM Buffers <12>, to the Scratchpad RAM <12> and Shared RAM <12> by performing a "walking one's" test on three memory locations. Address Decoder U610 <12> and Bus Arbiter <13> are also partially verified.

Description

1. Initialize the test locations.
 - Save the byte at location 4000hex.
 - Write the pattern FFhex to address 0000hex, 0001hex, and 4000hex.
2. Verify Address Decoder U610 <12>.
 - Read address 0000hex and verify that it is FFhex.
3. Perform a "walking one's" test on Shared RAM <12> address 0000hex. Terminate test if any verify operation fails.
 - Write the pattern 80hex to address 0000hex. Read the same address and verify that it was 80hex. Continue this write/read/verify sequence with the patterns 40hex, 20hex, 10hex, 08hex, 04hex, 02hex, 01hex.
4. Verify Address Decoder U610 <12>.
 - Write 00hex to address 0000hex.
 - Read address 0001hex and verify that it is FFhex.
5. Perform a "walking one's" test on Shared RAM <12> address 0001hex. Terminate test if any verify operation fails.
 - Write the pattern 80hex to address 0001hex. Read the same address and verify that it was 80hex. Continue this write/read/verify sequence with the patterns 40hex, 20hex, 10hex, 08hex, 04hex, 02hex, 01hex.
6. Verify Address Decoder U610 <12>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Memory	RAM Data (m[1-2]13X)
Acq[1-8]	Memory	RAM Data (a[1-8]13X)
Acq[1-8]	Memory	RAM Data (b[1-8]13X)
Acq[1-8]	Memory	RAM Data (c[1-8]13X)
Acq[1-8]	Memory	RAM Data (d[1-8]13X)

- Write 00hex to address 0001hex.
 - Read address 4000hex and verify that it is FFhex.
7. Perform a "walking one's" test on Scratchpad RAM <12> address 4000hex. Terminate test if any verify operation fails.
 - Write the pattern 80hex to address 4000hex. Read the same address and verify that it was 80hex. Continue this write/read/verify sequence with the patterns 40hex, 20hex, 10hex, 08hex, 04hex, 02hex, 01hex.
 8. Verify Address Decoder U610 <12>.
 - Write 00hex to address 4000hex.
 - Read address 0000hex, 0001hex, and 4000hex and verify that they are 00hex.
 9. Restore the saved byte to location 4000hex.

Error Index [m | a-d][1-8]131

One of the patterns read from Shared RAM or Scratchpad RAM <12> was not the pattern written. The following table shows the correspondence of the address to the Shared RAM or Scratchpad RAM <12> device:

<u>Address(hex)</u>	<u>Device</u>
0000	U620
0001	U720
4000	U710

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Memory	RAM Address (m[1-2]14X)
Acq[1-8]	Memory	RAM Address (a[1-8]14X)
Acq[1-8]	Memory	RAM Address (b[1-8]14X)
Acq[1-8]	Memory	RAM Address (c[1-8]14X)
Acq[1-8]	Memory	RAM Address (d[1-8]14X)

Routine Name RAM Address

Overview This test verifies the data and address lines from MPU U401 <12>, through MPU & Bus Buffers <12> and Shared RAM Buffers <12>, to the Scratchpad RAM <12> and Shared RAM <12> by performing a RAM test on both sections of memory. Address Decoder U610 <12> and Bus Arbiter <13> are also partially verified.

Description

1. Verify Shared RAM <12> address range 0000hex to 27FFhex. Terminate test if any verify operation fails.
 - Fill address range 0000hex to 27FFhex with the pattern AAhex.
 - Read and verify address 0000hex for AAhex. If so, write 55hex to address 0000hex. Increment the address and continue this read/verify/write sequence until address 27FFhex is reached.
 - Repeat the read/verify/write sequence, starting again at address 0000hex, for FFhex and 00hex (i.e., reading and verifying the previous pattern written and then writing the next pattern).
 - Make one last read/verify pass for the pattern 00hex starting at address 0000hex.
2. Save Shared RAM <12> and Scratchpad RAM <12> data in address range 2800hex to 4FFFhex by copying it to address range 0000hex to 27FFhex.
3. Verify Shared RAM <12> and Scratchpad RAM <12> address range 2800hex to 4FFFhex.
 - Perform the same procedures as in step 1.
5. Copy the saved Shared RAM <12> and Scratchpad RAM <12> data back from address range 0000hex to 27FFhex to address range 2800hex to 4FFFhex.
6. Save the Scratchpad RAM <12> data in address range 5000hex to 5FFFhex to Shared RAM <12> address range 0000hex to 0FFFhex.
7. Verify Scratchpad RAM <12> address range 5000hex to 5FFFhex.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Memory	RAM Address (m[1-2]14X)
Acq[1-8]	Memory	RAM Address (a[1-8]14X)
Acq[1-8]	Memory	RAM Address (b[1-8]14X)
Acq[1-8]	Memory	RAM Address (c[1-8]14X)
Acq[1-8]	Memory	RAM Address (d[1-8]14X)

- Perform the same procedures as in step 1.

8. Copy the saved Scratchpad RAM <12> data back from address range 0000hex to 0FFFhex to Shared RAM <12> address range 5000hex to 5FFFhex.

Error Index

[m | a-d][1-8]141

The pattern read from the displayed memory location in the Scratchpad RAM <12> or Shared RAM <12> was not the pattern written. The following table shows the correspondence of the address to the Shared RAM or Scratchpad RAM <12> device:

<u>Address(hex)</u>	<u>Device</u>
0000 - 3FFF	
XXX0	U620
XXX2	U620
XXX4	U620
XXX6	U620
XXX8	U620
XXXA	U620
XXXC	U620
XXXE	U620
XXX1	U720
XXX3	U720
XXX5	U720
XXX7	U720
XXX9	U720
XXXB	U720
XXXD	U720
XXXF	U720
4000 - 5FFF	U710

X - Don't Care

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Interrupt	Timer (m[1-2]21X)
Acq[1-8]	Interrupt	Timer (a[1-8]21X)
Acq[1-8]	Interrupt	Timer (b[1-8]21X)
Acq[1-8]	Interrupt	Timer (c[1-8]21X)
Acq[1-8]	Interrupt	Timer (d[1-8]21X)

Routine Name Timer

Overview This test verifies that FIRQ Timer U120 <12> and INTERRUPT CONTROL of MPU U401 <12> is functioning correctly. This is accomplished by counting the number of interrupts during a given amount of time and verifying that the count is within acceptable limits.

Description

1. Set up the processor to test the FIRQ interrupt.
 - Save the current FIRQ interrupt vector.
 - Load the interrupt vector for the FIRQ test interrupt handler.
 - Enable FIRQ interrupt.
2. Synchronize the processor to the FIRQ interrupt.
 - Wait for an interrupt to occur.
 - Clear the software timer/counter.
 - Wait for an interrupt to occur.
3. Increment the software timer/counter until three interrupts have occurred.
4. Disable FIRQ interrupt.
5. Verify that the software timer/counter is within the expected range.

Error Index
[m | a-d][1-8]211

The FIRQ Timer <12> interrupts did not occur at the proper rate.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Control	Dig Cntrl (m[1-2]31X)
Acq[1-8]	Control	Dig Cntrl (a[1-8]31X)
Acq[1-8]	Control	Dig Cntrl (b[1-8]31X)
Acq[1-8]	Control	Dig Cntrl (c[1-8]31X)
Acq[1-8]	Control	Dig Cntrl (d[1-8]31X)

Routine Name Dig Cntrl (Digital Control)

Overview This test verifies Diagnostic Mux Control Latch U1401 <11>, Chan Mux Latch U1501 <9>, Measurement & A/D Strobe Generator U901 <13>, and Sampling Head Digital Controls U1500 <11> by using a "walking ones" type test. This test also provides stimulus for the write-only register Front Panel LEDS/Switches U100 <12>.

Description

1. Save the current contents of Chan Mux Latch U1501 <9>, Measurement & A/D Strobe Generator U901 <13>, and Sampling Head Digital Controls U1500 <11>.
2. Write the test pattern 01hex to Diagnostic Mux Control Latch U1401 <11>, Chan Mux Latch U1501 <9>, Measurement & A/D Strobe Generator U901 <13>, Sampling Head Digital Controls U1500 <11>, and Front Panel LEDS/Switches U100 <12>.
3. Read Diagnostic Mux Control Latch U1401 <11>, Chan Mux Latch U1501 <9>, Measurement & A/D Strobe Generator U901 <13>, and Sampling Head Digital Controls U1500 <11> and verify that they contain the value 01hex.
4. Repeat steps 2 and 3 using the patterns: 02hex, 04hex, 08hex, 10hex, 20hex, 40hex, 80hex.
5. Restore the saved contents to Chan Mux Latch U1501 <9>, Measurement & A/D Strobe Generator U901 <13>, and Sampling Head Digital Controls U1500 <11>. Restore Front Panel LEDS/Switches U100 <12> to its previous state before the test.

Error Index
[m | a-d][1-8]311 The pattern read from Diagnostic Mux Control Latch U1401 <11> was not the pattern written.

Error Index
[m | a-d][1-8]312 The pattern read from Sampling Head Digital Controls U1500 <11> was not the pattern written.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Control	Dig Cntrl (m[1-2]31X)
Acq[1-8]	Control	Dig Cntrl (a[1-8]31X)
Acq[1-8]	Control	Dig Cntrl (b[1-8]31X)
Acq[1-8]	Control	Dig Cntrl (c[1-8]31X)
Acq[1-8]	Control	Dig Cntrl (d[1-8]31X)

Error Index

[m | a-d][1-8]313

The pattern read from Measurement & A/D Strobe Generator U901 <13> was not the pattern written.

Error Index

[m | a-d][1-8]314

The pattern read from Chan Mux Latch U1501 <9> was not the pattern written.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Diag ADC (m[1-2]41X)
Acq[1-8]	Signal Path	Diag ADC (a[1-8]41X)
Acq[1-8]	Signal Path	Diag ADC (b[1-8]41X)
Acq[1-8]	Signal Path	Diag ADC (c[1-8]41X)
Acq[1-8]	Signal Path	Diag ADC (d[1-8]41X)

Routine Name Diag ADC (Diagnostic A/D Converter)

Overview This test verifies Diagnostic A/D U1410 <11> by converting +1 and -1 volt reference signals and checking that the ADC codes are within acceptable ranges.

Description

1. Select +1 volt reference source.
 - Set Diagnostic Input Mux <11> U1411-1 low, U1411-16 high, and U1411-15 high to select U1411-10 +1V REF as input to the Diagnostic A/D.
2. Start the analog-to-digital conversion by reading and writing Diagnostic A/D U1410 <11>.
3. Read the ADC code from Diagnostic A/D U1410 <11>.
 - Wait until A/D Status Register U1300 <11> pin 2 is high.
 - Read Diagnostic A/D through buffer U1400.
4. Select -1 volt reference source.
 - Set Diagnostic Input Mux <11> U1411-1 high, U1411-16 high, and U1411-15 high to select U1411-9 -1V REF as input to the Diagnostic A/D.
5. Start the analog-to-digital conversion by reading and writing Diagnostic A/D U1410 <11>.
6. Read the ADC code from Diagnostic A/D U1410 <11>.
 - Wait until A/D Status Register U1300 <11> pin 2 is high.
 - Read Diagnostic A/D through buffer U1400.
7. Verify that the ADC codes read for +1 and -1 volts are within the expected ranges.

Error Index
[m | a-d][1-8]411

The ADC code for +1 volt was not within the acceptable range.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Diag ADC (m[1-2]41X)
Acq[1-8]	Signal Path	Diag ADC (a[1-8]41X)
Acq[1-8]	Signal Path	Diag ADC (b[1-8]41X)
Acq[1-8]	Signal Path	Diag ADC (c[1-8]41X)
Acq[1-8]	Signal Path	Diag ADC (d[1-8]41X)

Error Index	
[m a-d][1-8]412	The ADC code for -1 volt was not within the acceptable range.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Strobes (m[1-2]42X)
Acq[1-8]	Signal Path	Strobes (a[1-8]42X)
Acq[1-8]	Signal Path	Strobes (b[1-8]42X)
Acq[1-8]	Signal Path	Strobes (c[1-8]42X)
Acq[1-8]	Signal Path	Strobes (d[1-8]42X)

Routine Name Strobes

Overview This test partially verifies Interrupt Latch <13>, Measurement & A/D Strobe Generator <13>, Chan Input Mux A <9>, Chan Input Mux B <9>, Prog Gain Amp A <9>, Prog Gain Amp B <9>, Chan A/D A <9>, and Chan A/D B <9>. This is done by connecting a ground reference signal to the input of each programmable amplifier, minimizing each channel's offsets, applying software generated strobes to each channel's A/D, and then checking the ADC codes read from each channel.

Description

1. Configure Chan A/D A <9> and Chan A/D B <9> inputs.
 - Select the Chan Input Mux A U301-1 <9> and Chan Input Mux B U100-1 <9> ground reference signals by setting pins 13, 14, and 15 on each device low.
 - Set x2/x20 gains to x2 and variable gains to one by setting Chan Mux Latch U1501-12,13 high <9> and writing 155_{hex} to both halves of Prog Gain Amp Gain Control Voltage U1201 <10>.
 - Set the offset near ground by writing 2000_{hex} to Offset Control Voltage U811 <10> and Offset Control Voltage U810 <10>, and by writing 80_{hex} to Offset Correction and TDR Delay Control Voltage U1200 (parts a & b) <10>.
 - Enable Chan A/D A <9> and Chan A/D B <9> by setting Measurement & A/D Strobe Generator U901 <13> pins 16 & 17 high.
 - Enable the processor-generated strobe (MPUSTB) by setting Enable Register U801-18 MSTREN(L) <13> low.
2. Initialize the waveform address counter.
 - Pulse Hardware Measurement System U820-13 <13> WACCLR(L).
 - Pulse Hardware Measurement System U820-11 <13> WACCLK eight times.
 - Pulse Hardware Measurement System U820-13 <13> WACCLR(L).

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Strobes (m[1-2]42X)
Acq[1-8]	Signal Path	Strobes (a[1-8]42X)
Acq[1-8]	Signal Path	Strobes (b[1-8]42X)
Acq[1-8]	Signal Path	Strobes (c[1-8]42X)
Acq[1-8]	Signal Path	Strobes (d[1-8]42X)

3. Initialize Shared Ram <12>.
 - Write the pattern *DEBC_{hex}* to memory starting at address *0002_{hex}* through address *2802_{hex}*.
4. Verify that Shared Ram <12> has not changed (i.e. that the waveform address counter is counting correctly).
 - Read memory address *0002_{hex}* and verify that it is still *DEBC_{hex}*.
5. Generate a software strobe.
 - Pulse Interrupt Latch U1100B-5 MPUSTB <13>.
6. Verify that Shared Ram <12> has changed.
 - Read memory address *0002_{hex}* and verify that the low byte is between *60_{hex}* and *A0_{hex}* and the high byte is between *60_{hex}* and *A0_{hex}*.
7. Increment the address to *0004_{hex}* and repeat steps 4 - 6 for memory addresses *0004_{hex}* to *2802_{hex}*.

Error Index

[m | a-d][1-8]421

The waveform address counter is either ahead or behind one or more counts.

Error Index

[m | a-d][1-8]422

The ADC codes were found to be out of the acceptable range.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Gain (m[1-2]43X)
Acq[1-8]	Signal Path	Gain (a[1-8]43X)
Acq[1-8]	Signal Path	Gain (b[1-8]43X)
Acq[1-8]	Signal Path	Gain (c[1-8]43X)
Acq[1-8]	Signal Path	Gain (d[1-8]43X)

Routine Name Gain

Overview This test verifies that **Prog Gain Amp Gain Control Voltage** <10> functions correctly by using a "walking ones" type test on DAC U1201.

Description

1. Write *10hex* to **Diagnostic Mux Control Latch U1401** <11> to select **Diagnostic Input Mux U1330-4 VGAINA** <11> as the input to **Diagnostic A/D** <11>.
2. Write the test pattern *0Ehex* to **Prog Gain Amp Gain Control Voltage** part a <10>. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing **Diagnostic A/D U1410** <11>.
4. Read the ADC code from **Diagnostic A/D U1410** <11>.
 - Wait until **A/D Status Register U1300** <11> pin 2 is high.
 - Read **Diagnostic A/D** through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns *01Bhex*, *036hex*, *06Bhex*, *0B6hex*, *1B6hex*, *356hex*, *6ABhex*.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write *11hex* to **Diagnostic Mux Control Latch U1401** <11> to select **Diagnostic Input Mux U1330-5 VGAINB** <11> as the input to **Diagnostic A/D** <11>.
8. Repeat steps 2 through 6 using **Prog Gain Amp Gain Control Voltage** part b <10> instead of **Prog Gain Amp Gain Control Voltage** part a <10>.

Error Index
[m | a-d][1-8]431

The **Prog Gain Amp Gain Control Voltage** DAC U1201 part a <10> did not function as expected.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Gain (m[1-2]43X)
Acq[1-8]	Signal Path	Gain (a[1-8]43X)
Acq[1-8]	Signal Path	Gain (b[1-8]43X)
Acq[1-8]	Signal Path	Gain (c[1-8]43X)
Acq[1-8]	Signal Path	Gain (d[1-8]43X)

Error Index

[m | a-d][1-8]432

The Prog Gain Amp Gain Control Voltage DAC U1201 part b <10> did not function as expected.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Offset (m[1-2]44X)
Acq[1-8]	Signal Path	Offset (a[1-8]44X)
Acq[1-8]	Signal Path	Offset (b[1-8]44X)
Acq[1-8]	Signal Path	Offset (c[1-8]44X)
Acq[1-8]	Signal Path	Offset (d[1-8]44X)

Routine Name Offset

Overview This test verifies that **Offset Control Voltage** <10> functions correctly by using "walking ones" type tests on DACs U810 and U811.

Description

1. Write *2Ahex* to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1230-11 VOSA <11> as the input to **Diagnostic A/D** <11>.
2. Write the test pattern *0040hex* to **Offset Control Voltage** U811 <10> twice. The first write loads the high byte, the second write loads the low byte. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing **Diagnostic A/D** U1410 <11>.
4. Read the ADC code from **Diagnostic A/D** U1410 <11>.
 - Wait until **A/D Status Register** U1300 <11> pin 2 is high.
 - Read **Diagnostic A/D** through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns *0080hex*, *0100hex*, *0200hex*, *0400hex*, *0800hex*, *1000hex*, *2000hex*.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write *2Bhex* to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1230-10 VOSB <11> as the input to **Diagnostic A/D** <11>.
8. Repeat steps 2 through 6 using **Offset Control Voltage** DAC U810 <10> instead of **Offset Control Voltage** DAC U811 <10>.

Error Index
[m | a-d][1-8]441 The **Offset Control Voltage** DAC U811 <10> did not function as expected.

Error Index
[m | a-d][1-8]442 The **Offset Control Voltage** DAC U810 <10> did not function as expected.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Correction (m[1-2]45X)
Acq[1-8]	Signal Path	Correction (a[1-8]45X)
Acq[1-8]	Signal Path	Correction (b[1-8]45X)
Acq[1-8]	Signal Path	Correction (c[1-8]45X)
Acq[1-8]	Signal Path	Correction (d[1-8]45X)

Routine Name Correction

Overview This test verifies that Offset Correction & TDR Delay Control Voltage <10> functions correctly by using "walking ones" type tests on DAC U1200 parts a and b.

Description

1. Write *18hex* to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1330-13 OSCORA <11> as the input to Diagnostic A/D <11>.
2. Write the test pattern *01hex* to Offset Correction & TDR Delay Control Voltage part a <10>. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing Diagnostic A/D U1410 <11>.
4. Read the ADC code from Diagnostic A/D U1410 <11>.
 - Wait until A/D Status Register U1300 <11> pin 2 is high.
 - Read Diagnostic A/D through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns *02hex*, *04hex*, *08hex*, *10hex*, *20hex*, *40hex*, *80hex*.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write *18hex* to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1330-12 OSCORB <11> as the input to Diagnostic A/D <11>.
8. Repeat steps 2 through 6 using Offset Correction & TDR Delay Control Voltage DAC U1200 part b <10> instead of Offset Correction & TDR Delay Control Voltage DAC U1200 part a <10>.

Error Index
[m | a-d][1-8]451 The Offset Correction & TDR Delay Control Voltage DAC U1200 part a <10> did not function as expected.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Correction (m[1-2]45X)
Acq[1-8]	Signal Path	Correction (a[1-8]45X)
Acq[1-8]	Signal Path	Correction (b[1-8]45X)
Acq[1-8]	Signal Path	Correction (c[1-8]45X)
Acq[1-8]	Signal Path	Correction (d[1-8]45X)

Error Index

[m | a-d][1-8]452

The Offset Correction & TDR Delay Control Voltage DAC U1200 part b <10> did not function as expected.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Sig Path A (m[1-2]46X)
Acq[1-8]	Signal Path	Sig Path A (a[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (b[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (c[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (d[1-8]46X)

Routine Name Sig Path A (Signal Path A)

Overview This test verifies that **Prog Gain Amp A** <9> responds correctly to all of the following control inputs:

- **Chan Input Mux A** <9> inputs of ground, +16mv, +2 volts, and -2 volts are connected to the amplifier, digitized, and then verified.
- **Offset Correction & TDR Delay Control Voltage DAC U1200** part a <10> and **Offset Control Voltage DAC U811** <10> are checked by setting them at each end of their ranges and verifying that the changes are acceptable.
- The variable gain of **Prog Gain Amp Gain Control Voltage DAC U1201** part a <10> is checked in the following manner: set the variable gain to x1, adjust the offset control to +0.1 volts and -0.1 volts, and then measure the range due to the offset control. Next, change the variable gain to x2.5 and repeat the range measurement for the offset control. Finally, verify the ratio of measured ranges at both variable gain settings.
- The x2/x20 switch, **Prog Gain Amp A U430** <9>, is checked in the following manner: set the x2/x20 switch to x2; set the offset control to +0.25 volts; verify that the ADC value is above ground but not overranged; switch to x20; verify that the ADC value is above ground and is overranged. Set the x2/x20 switch back to x2; set the offset control to -0.25 volts; verify that the ADC value is below ground but not underranged; switch to x20; verify that the ADC value is below ground and is underranged.

Description 1. Configure **Prog Gain Amp A** <9>.

- Select the **Chan Input Mux A U301-1** <9> ground reference signal by setting pins 13, 14, and 15 low.
- Set x2/x20 gain to x2 by setting **Chan Mux Latch U1501-13** <9> high.
- Set variable gain to x1 by writing 155_{hex} to **Prog Gain Amp Gain Control Voltage DAC U1201** part a <10>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Sig Path A (m[1-2]46X)
Acq[1-8]	Signal Path	Sig Path A (a[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (b[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (c[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (d[1-8]46X)

- Set the offset near ground by writing `2000hex` to Offset Control Voltage DAC U811 <10>.
 - Enable Chan A/D A <9> by setting Measurement & A/D Strobe Generator U901-16 <13> high.
 - Enable the processor-generated strobe (MPUSTB) by setting Enable Register U801-18 MSTREN(L) <13> low.
2. Check the ground reference signal.
- Clear the waveform address counter by pulsing Hardware Measurement System U820-13 WACCLR(L) <13>.
 - Acquire four data values by pulsing Interrupt Latch U1100B-5 MPUSTB <13> four times.
 - Average the data values in memory locations `0003hex`, `0005hex`, `0007hex`, and `0009hex`. Save this value for step 3.
 - Verify that the average is between `60hex` and `A0hex`. If not, then terminate the test.
3. Check the +16mv reference signal.
- Select the Chan Input Mux A U301-2 +16mv <9> reference signal by setting pin 13 high and pins 14 & 15 low.
 - Perform the first three steps listed under step 2 and verify that this averaged value is within the following range (else terminate the test):
- $$\text{Step 2 value} + 1 \leq \text{average} \leq \text{Step 2 value} + 3$$
4. Check the +2 volt reference signal.
- Select the Chan Input Mux A U301-4 +2V REF <9> reference signal by setting pins 13 & 14 high and pin 15 low.
 - Perform the first three steps listed under step 2 and verify that the averaged value is `FFhex` (i.e. overrange). If not, terminate the test.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Sig Path A (m[1-2]46X)
Acq[1-8]	Signal Path	Sig Path A (a[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (b[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (c[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (d[1-8]46X)

5. Check the -2 volt reference signal.
 - Select the **Chan Input Mux A U301-3 -2V REF <9>** reference signal by setting pins 13 low, 14 high, and 15 low.
 - Perform the first three steps listed under step 2 and verify that the averaged value is *00hex* (i.e. underrange). If not, terminate the test.
 - Select the **Chan Input Mux A U301-1 <9>** ground reference signal by setting pins 13, 14, and 15 low.
6. Check the offset correction range.
 - Set **Offset Correction & TDR Delay Control Voltage OSCORA <10>** to +256mv by writing *00hex* to DAC U1200 part a <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set **Offset Correction & TDR Delay Control Voltage OSCORA <10>** to -256mv by writing *FFhex* to DAC U1200 part a <10>.
 - Perform the first three steps listed under step 2 and verify that the difference between the average taken earlier in this step and the average just taken is between *58hex* and *84hex*, inclusive. If not, terminate the test.
 - Set **Offset Correction & TDR Delay Control Voltage OSCORA <10>** to 0 volts by writing *80hex* to DAC U1200 part a <10>.
7. Check the offset control range.
 - Set **Offset Control Voltage VOSA <10>** to +0.5 volts by writing *2800hex* to DAC U811 <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set **Offset Control Voltage VOSA <10>** to -0.5 volts by writing *1800hex* to DAC U811 <10>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Sig Path A (m[1-2]46X)
Acq[1-8]	Signal Path	Sig Path A (a[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (b[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (c[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (d[1-8]46X)

- Perform the first three steps listed under step 2 and verify that the difference between the average taken earlier in this step and the average just taken is between 67_{hex} and 99_{hex} , inclusive. If not, terminate the test.
8. Check the variable gain.
- Set **Offset Control Voltage** VOSA <10> to +0.1 volts by writing 2186_{hex} to DAC U811 <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set **Offset Control Voltage** VOSA <10> to -0.1 volts by writing $1E7A_{hex}$ to DAC U811 <10>.
 - Perform the first three steps listed under step 2 and subtract this average from the previous average in this step. For later use, call this the x1 variable gain range.
 - Set variable gain to x2.5 by writing $6A9_{hex}$ to Prog Gain Amp Gain Control Voltage DAC U1201 part a <10>.
 - Set **Offset Control Voltage** VOSA <10> to +0.1 volts by writing 2186_{hex} to DAC U811 <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set **Offset Control Voltage** VOSA <10> to -0.1 volts by writing $1E7A_{hex}$ to DAC U811 <10>.
 - Perform the first three steps listed under step 2 and subtract this average from the previous average in this step. Call this the x2.5 variable gain range.
 - Verify that the ratio between the x2.5 and x1 variable gain range is between 4 and 6, inclusive (the programmable gain amplifier is in x2 mode, so the ratio should effectively be $2.5 * 2$, i.e. 5). If not, terminate the test.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Sig Path A (m[1-2]46X)
Acq[1-8]	Signal Path	Sig Path A (a[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (b[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (c[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (d[1-8]46X)

- Set variable gain to x1 by writing 155_{hex} to Prog Gain Amp Gain Control Voltage DAC U1201 part a <10>.
9. Check the x2/x20 switch.
- Clear the waveform address counter by pulsing Hardware Measurement System U820-13 WACCLR(L) <13>.
 - Set x2/x20 gain to x2 by setting Chan Mux Latch U1501-13 <9> high.
 - Set Offset Control Voltage VOSA <10> to +0.25 volts by writing 23CF_{hex} to DAC U811 <10>.
 - Acquire one data value by pulsing Interrupt Latch U1100B-5 MPUSTB <13> once.
 - Set x2/x20 gain to x20 by setting Chan Mux Latch U1501-13 <9> low.
 - Acquire another data value by pulsing Interrupt Latch U1100B-5 MPUSTB <13> once.
 - Verify that the first acquired data byte, in memory location 0003_{hex}, is between 80_{hex} and FF_{hex}, inclusive.
 - Verify that the second acquired data byte, in memory location 0005_{hex}, is FF_{hex} (i.e. overrange).
 - Clear the waveform address counter by pulsing Hardware Measurement System U820-13 WACCLR(L) <13>.
 - Set x2/x20 gain to x2 by setting Chan Mux Latch U1501-13 <9> high.
 - Set Offset Control Voltage VOSA <10> to -0.25 volts by writing 1C31_{hex} to DAC U811 <10>.
 - Acquire one data value by pulsing Interrupt Latch U1100B-5 MPUSTB <13> once.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Sig Path A (m[1-2]46X)
Acq[1-8]	Signal Path	Sig Path A (a[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (b[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (c[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (d[1-8]46X)

- Set x2/x20 gain to x20 by setting Chan Mux Latch U1501-13 <9> low.
- Acquire another data value by pulsing Interrupt Latch U1100B-5 MPUSTB <13> once.
- Verify that the first acquired data byte, in memory location 0003hex, is between 00hex and 80hex, inclusive.
- Verify that the second acquired data byte, in memory location 0005hex, is 00hex (i.e. underrange).
- Repeat this step (step 9) three more times.

Error Index
[m | a-d][1-8]461

The ADC code read from Chan A/D A <9>, when the input was connected to the ground reference signal, was not between 60hex and A0hex.

Error Index
[m | a-d][1-8]462

The ADC code read from Chan A/D A <9>, when the input was connected to the +16mv reference signal, was outside of the acceptable range.

Error Index
[m | a-d][1-8]463

The ADC code read from Chan A/D A <9>, when the input was connected to the +2V REF reference signal, was not FFhex (i.e. overrange).

Error Index
[m | a-d][1-8]464

The ADC code read from Chan A/D A <9>, when the input was connected to the -2V REF reference signal, was not 00hex (i.e. underrange).

Error Index
[m | a-d][1-8]465

The offset correction range, from Offset Correction & TDR Delay Control Voltage DAC U1200 part a <10>, was either too small or too large.

Error Index
[m | a-d][1-8]466

The offset control range, from Offset Control Voltage DAC U811 <10>, was either too small or too large.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Sig Path A (m[1-2]46X)
Acq[1-8]	Signal Path	Sig Path A (a[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (b[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (c[1-8]46X)
Acq[1-8]	Signal Path	Sig Path A (d[1-8]46X)

Error Index [m a-d][1-8]467	The ratio of the x2.5 to x1 variable gain was not within the expected range.
----------------------------------	--

Error Index [m a-d][1-8]468	One of the ADC codes read from Chan A/D A <9>, when the x2/x20 switch was being tested above ground, was not within the expected range.
----------------------------------	---

Error Index [m a-d][1-8]469	One of the ADC codes read from Chan A/D A <9>, when the x2/x20 switch was being tested below ground, was not within the expected range.
----------------------------------	---

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Sig Path B (m[1-2]47X)
Acq[1-8]	Signal Path	Sig Path B (a[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (b[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (c[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (d[1-8]47X)

Routine Name Sig Path B (Signal Path B)

Overview This test verifies that **Prog Gain Amp B <9>** responds correctly to all of the following control inputs:

- **Chan Input Mux B <9>** inputs of ground, +16mv, +2 volts, and -2 volts are connected to the amplifier, digitized, and then verified.
- **Offset Correction & TDR Delay Control Voltage DAC U1200 part b <10>** and **Offset Control Voltage DAC U810 <10>** are checked by setting them at each end of their ranges and verifying that the changes are acceptable.
- The variable gain of **Prog Gain Amp Gain Control Voltage DAC U1201 part b <10>** is checked in the following manner: set the variable gain to x1, adjust the offset control to +0.1 volts and -0.1 volts, and then measure the range due to the offset control. Next, change the variable gain to x2.5 and repeat the range measurement for the offset control. Finally, verify the ratio of measured ranges at both variable gain settings.
- The x2/x20 switch, **Prog Gain Amp B U230 <9>**, is checked in the following manner: set the x2/x20 switch to x2; set the offset control to +0.25 volts; verify that the ADC value is above ground but not overranged; switch to x20; verify that the ADC value is above ground and is overranged. Set the x2/x20 switch back to x2; set the offset control to -0.25 volts; verify that the ADC value is below ground but not underranged; switch to x20; verify that the ADC value is below ground and is underranged.

Description

1. Configure **Prog Gain Amp B <9>**.
 - Select the **Chan Input Mux B U100-1 <9>** ground reference signal by setting pins 13, 14, and 15 low.
 - Set x2/x20 gain to x2 by setting **Chan Mux Latch U1501-12 <9>** high.
 - Set variable gain to x1 by writing 155_{hex} to **Prog Gain Amp Gain Control Voltage DAC U1201 part b <10>**.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Sig Path B (m[1-2]47X)
Acq[1-8]	Signal Path	Sig Path B (a[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (b[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (c[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (d[1-8]47X)

- Set the offset near ground by writing `2000hex` to Offset Control Voltage DAC U810 <10>.
 - Enable Chan A/D B <9> by setting Measurement & A/D Strobe Generator U901-17 <13> high.
 - Enable the processor-generated strobe (MPUSTB) by setting Enable Register U801-18 MSTREN(L) <13> low.
2. Check the ground reference signal.
- Clear the waveform address counter by pulsing Hardware Measurement System U820-13 WACCLR(L) <13>.
 - Acquire four data values by pulsing Interrupt Latch U1100B-5 MPUSTB <13> four times.
 - Average the data values in memory locations `0002hex`, `0004hex`, `0006hex`, and `0008hex`. Save this value for step 3.
 - Verify that the average is between `60hex` and `A0hex`. If not, then terminate the test.
3. Check the +16mv reference signal.
- Select the Chan Input Mux B U100-2 +16mv <9> reference signal by setting pin 13 high and pins 14 & 15 low.
 - Perform the first three steps listed under step 2 and verify that this averaged value is within the following range (else terminate the test):
- $$\text{Step 2 value} + 1 \leq \text{average} \leq \text{Step 2 value} + 3$$
4. Check the +2 volt reference signal.
- Select the Chan Input Mux B U100-4 +2V REF <9> reference signal by setting pins 13 & 14 high and pin 15 low.
 - Perform the first three steps listed under step 2 and verify that the averaged value is `FFhex` (i.e. overrange). If not, terminate the test.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Sig Path B (m[1-2]47X)
Acq[1-8]	Signal Path	Sig Path B (a[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (b[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (c[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (d[1-8]47X)

5. Check the -2 volt reference signal.
 - Select the **Chan Input Mux B** U100-3 -2V REF <9> reference signal by setting pins 13 low, 14 high, and 15 low.
 - Perform the first three steps listed under step 2 and verify that the averaged value is 00hex (i.e. underrange). If not, terminate the test.
 - Select the **Chan Input Mux B** U100-1 <9> ground reference signal by setting pins 13, 14, and 15 low.
6. Check the offset correction range.
 - Set **Offset Correction & TDR Delay Control Voltage** OSCORB <10> to +256mv by writing 00hex to DAC U1200 part b <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set **Offset Correction & TDR Delay Control Voltage** OSCORB <10> to -256mv by writing FFhex to DAC U1200 part b <10>.
 - Perform the first three steps listed under step 2 and verify that the difference between the average taken earlier in this step and the average just taken is between 58hex and 84hex, inclusive. If not, terminate the test.
 - Set **Offset Correction & TDR Delay Control Voltage** OSCORB <10> to 0 volts by writing 80hex to DAC U1200 part b <10>.
7. Check the offset control range.
 - Set **Offset Control Voltage** VOSB <10> to +0.5 volts by writing 2800hex to DAC U810 <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set **Offset Control Voltage** VOSB <10> to -0.5 volts by writing 1800hex to DAC U810 <10>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Sig Path B (m[1-2]47X)
Acq[1-8]	Signal Path	Sig Path B (a[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (b[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (c[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (d[1-8]47X)

- Perform the first three steps listed under step 2 and verify that the difference between the average taken earlier in this step and the average just taken is between 67_{hex} and 99_{hex} , inclusive. If not, terminate the test.
8. Check the variable gain.
- Set Offset Control Voltage VOSB <10> to +0.1 volts by writing 2186_{hex} to DAC U810 <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set Offset Control Voltage VOSB <10> to -0.1 volts by writing $1E7A_{hex}$ to DAC U810 <10>.
 - Perform the first three steps listed under step 2 and subtract this average from the previous average in this step. For later use, call this the x1 variable gain range.
 - Set variable gain to x2.5 by writing $6A9_{hex}$ to Prog Gain Amp Gain Control Voltage DAC U1201 part b <10>.
 - Set Offset Control Voltage VOSB <10> to +0.1 volts by writing 2186_{hex} to DAC U810 <10>.
 - Perform the first three steps listed under step 2 and retain this averaged value for later use.
 - Set Offset Control Voltage VOSB <10> to -0.1 volts by writing $1E7A_{hex}$ to DAC U810 <10>.
 - Perform the first three steps listed under step 2 and subtract this average from the previous average in this step. Call this the x2.5 variable gain range.
 - Verify that the ratio between the x2.5 and x1 variable gain range is between 4 and 6, inclusive (the programmable gain amplifier is in x2 mode, so the ratio should effectively be $2.5 * 2$, i.e. 5). If not, terminate the test.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Sig Path B (m[1-2]47X)
Acq[1-8]	Signal Path	Sig Path B (a[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (b[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (c[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (d[1-8]47X)

- Set variable gain to x1 by writing 155_{hex} to Prog Gain Amp Gain Control Voltage DAC U1201 part b <10>.
9. Check the x2/x20 switch.
- Clear the waveform address counter by pulsing **Hardware Measurement System** U820-13 WACCLR(L) <13>.
 - Set x2/x20 gain to x2 by setting **Chan Mux Latch** U1501-12 <9> high.
 - Set **Offset Control Voltage** VOSB <10> to +0.25 volts by writing 23CF_{hex} to DAC U810 <10>.
 - Acquire one data value by pulsing **Interrupt Latch** U1100B-5 MPUSTB <13> once.
 - Set x2/x20 gain to x20 by setting **Chan Mux Latch** U1501-12 <9> low.
 - Acquire another data value by pulsing **Interrupt Latch** U1100B-5 MPUSTB <13> once.
 - Verify that the first acquired data byte, in memory location 0002_{hex}, is between 80_{hex} and FF_{hex}, inclusive.
 - Verify that the second acquired data byte, in memory location 0004_{hex}, is FF_{hex} (i.e. overrange).
 - Clear the waveform address counter by pulsing **Hardware Measurement System** U820-13 WACCLR(L) <13>.
 - Set x2/x20 gain to x2 by setting **Chan Mux Latch** U1501-12 <9> high.
 - Set **Offset Control Voltage** VOSB <10> to -0.25 volts by writing 1C31_{hex} to DAC U810 <10>.
 - Acquire one data value by pulsing **Interrupt Latch** U1100B-5 MPUSTB <13> once.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Signal Path	Sig Path B (m[1-2]47X)
Acq[1-8]	Signal Path	Sig Path B (a[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (b[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (c[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (d[1-8]47X)

- Set x2/x20 gain to x20 by setting Chan Mux Latch U1501-12 <9> low.
- Acquire another data value by pulsing Interrupt Latch U1100B-5 MPUSTB <13> once.
- Verify that the first acquired data byte, in memory location 0002hex, is between 00hex and 80hex, inclusive.
- Verify that the second acquired data byte, in memory location 0004hex, is 00hex (i.e. underrange).
- Repeat this step (step 9) three more times.

Error Index
[m | a-d][1-8]471

The ADC code read from Chan A/D B <9>, when the input was connected to the ground reference signal, was not between 60hex and A0hex.

Error Index
[m | a-d][1-8]472

The ADC code read from Chan A/D B <9>, when the input was connected to the +16mv reference signal, was outside of the acceptable range.

Error Index
[m | a-d][1-8]473

The ADC code read from Chan A/D B <9>, when the input was connected to the +2V REF reference signal, was not FFhex (i.e. overrange).

Error Index
[m | a-d][1-8]474

The ADC code read from Chan A/D B <9>, when the input was connected to the -2V REF reference signal, was not 00hex (i.e. underrange).

Error Index
[m | a-d][1-8]475

The offset correction range, from Offset Correction & TDR Delay Control Voltage DAC U1200 part b <10>, was either too small or too large.

Error Index
[m | a-d][1-8]476

The offset control range, from Offset Control Voltage DAC U810 <10>, was either too small or too large.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Signal Path	Sig Path B (m[1-2]47X)
Acq[1-8]	Signal Path	Sig Path B (a[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (b[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (c[1-8]47X)
Acq[1-8]	Signal Path	Sig Path B (d[1-8]47X)

Error Index
[m | a-d][1-8]477

The ratio of the x2.5 to x1 variable gain was not within the expected range.

Error Index
[m | a-d][1-8]478

One of the ADC codes read from Chan A/D B <9>, when the x2/x20 switch was being tested above ground, was not within the expected range.

Error Index
[m | a-d][1-8]479

One of the ADC codes read from Chan A/D B <9>, when the x2/x20 switch was being tested below ground, was not within the expected range.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Ref Level (m[1-2]51X)
Acq[1-8]	Measurement	Ref Level (a[1-8]51X)
Acq[1-8]	Measurement	Ref Level (b[1-8]51X)
Acq[1-8]	Measurement	Ref Level (c[1-8]51X)
Acq[1-8]	Measurement	Ref Level (d[1-8]51X)

Routine Name Ref Level (Reference Level)

Overview This test verifies that **Comparator Control Voltage** <10> functions correctly by using "walking ones" type tests on DACs U711 and U710.

Description

1. Write 28_{hex} to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1230-13 VCREFA <11> as the input to **Diagnostic A/D** <11>.
2. Write the test pattern 0040_{hex} to **Comparator Control Voltage** U711 <10> twice. The first write loads the high byte, the second write loads the low byte. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing **Diagnostic A/D** U1410 <11>.
4. Read the ADC code from **Diagnostic A/D** U1410 <11>.
 - Wait until **A/D Status Register** U1300 <11> pin 2 is high.
 - Read **Diagnostic A/D** through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns 0080_{hex}, 0100_{hex}, 0200_{hex}, 0400_{hex}, 0800_{hex}, 1000_{hex}, 2000_{hex}.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write 29_{hex} to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1230-12 VCREFB <11> as the input to **Diagnostic A/D** <11>.
8. Repeat steps 2 through 6 using **Comparator Control Voltage** DAC U710 <10> instead of **Comparator Control Voltage** DAC U711 <10>.

Error Index
[m | a-d][1-8]511 The **Comparator Control Voltage** DAC U711 <10> did not function as expected.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Ref Level (m[1-2]51X)
Acq[1-8]	Measurement	Ref Level (a[1-8]51X)
Acq[1-8]	Measurement	Ref Level (b[1-8]51X)
Acq[1-8]	Measurement	Ref Level (c[1-8]51X)
Acq[1-8]	Measurement	Ref Level (d[1-8]51X)

Error Index
[m | a-d][1-8]512

The Comparator Control Voltage DAC U710 <10> did not function as expected.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

Routine Name Filter A

Overview This test verifies **A Meas Comparator <9>** (by using three voltage reference inputs) and the Dot Filter, Transition Found, and Dot Found capabilities of **Hardware Measurement System LCA U820 <13>** channel A. All of this is accomplished by configuring the channel A measurement system, simulating hardware strobes, and checking the results read back through **A/D Status Register <13>**.

Description

1. Configure hardware measurement channel A.
 - Set **Measurement & A/D Strobe Generator U901 <13>** pins 16 & 17 low to disable **Chan A/D A** and **Chan A/D B <9>**.
 - Set **Measurement & A/D Strobe Generator U901 <13>** pins 14 & 15 low to disable the **Hardware Measurement System LCA U820 Dot Filters** (via disabling **U820-31 DFCLKA** and **U820-37 DFCLKB**).
 - Set **Measurement & A/D Strobe Generator U901 <13>** pins 12 & 13 high.
 - Set **Measurement & A/D Strobe Generator U901-18 <13>** high to enable **A Meas Comparator <9>**.
 - Set **Measurement & A/D Strobe Generator U901-19 <13>** low to disable **B Meas Comparator <9>**.
2. Verify that **A Meas Comparator <9>** correctly sees the ground voltage reference.
 - Enable the processor-generated strobe (**MPUSTB**) by setting **Enable Register U801-18 MSTREN(L) <13>** low.
 - Set **Chan Mux Latch U1501 <9>** pins 14, 15, and 16 low to select the ground voltage reference at **Chan Input Mux A U301-1 <9>**.
 - Write **1E7A_{hex}** to **Comparator Control Voltage DAC U711 <10>** and wait a short amount of time for the DAC to settle.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

- Pulse A Meas Comparator U400-6 CMPSTBA <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U400.
 - Verify that data bit 0 (COMPA) of A/D Status Register <13> is low.
 - Write 2186_{hex} to Comparator Control Voltage DAC U711 <10> and wait a short amount of time for the DAC to settle.
 - Pulse A Meas Comparator U400-6 CMPSTBA <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U400.
 - Verify that data bit 0 (COMPA) of A/D Status Register <13> is high.
3. Verify that A Meas Comparator <9> correctly sees the -2 volt reference.
- Set Chan Mux Latch U1501 <9> pins 14 & 16 low and pin 15 high to select -2V REF at Chan Input Mux A U301-3 <9>.
 - Write 0000_{hex} to Comparator Control Voltage DAC U711 <10> and wait a short amount of time for the DAC to settle.
 - Pulse A Meas Comparator U400-6 CMPSTBA <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U400.
 - Verify that data bit 0 (COMPA) of A/D Status Register <13> is low.
 - Write 030_{hex} to Comparator Control Voltage DAC U711 <10> and wait a short amount of time for the DAC to settle.
 - Pulse A Meas Comparator U400-6 CMPSTBA <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U400.
 - Verify that data bit 0 (COMPA) of A/D Status Register <13> is high.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

4. Verify that A Meas Comparator <9> correctly sees the +2 volt reference.
 - Set Chan Mux Latch U1501 <9> pin 14 low and pins 15 & 16 high to select +2V REF at Chan Input Mux A U301-4 <9>.
 - Write 3FFF_{hex} to Comparator Control Voltage DAC U711 <10> and wait a short amount of time for the DAC to settle.
 - Pulse A Meas Comparator U400-6 CMPSTBA <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U400.
 - Verify that data bit 0 (COMPA) of A/D Status Register <13> is high.
 - Write 3CF4_{hex} to Comparator Control Voltage DAC U711 <10> and wait a short amount of time for the DAC to settle.
 - Pulse A Meas Comparator U400-6 CMPSTBA <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U400.
 - Verify that data bit 0 (COMPA) of A/D Status Register <13> is low.
5. Verify that the Channel A Dot Filter of Hardware Measurement System U820 <13> functions correctly.
 - Set Measurement & A/D Strobe Generator U901-14 <13> high to enable the Dot Filter Clock A (DFCLKA).
 - Set Chan Mux Latch U1501 <9> pins 14, 15, and 16 low to select the ground voltage reference at Chan Input Mux A U301-1 <9>.
 - Load one of the Hardware Measurement System LCA U820 <13> input latches with 6_{hex} (for the dot filter counter).
 - Write 0000_{hex} to Comparator Control Voltage DAC U711 <10> to set Hardware Measurement System U820-22 COMPA <13> high (via A Meas Comparator U400-7 COMPA <9>).

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

- Pulse **Hardware Measurement System U820-2 DLACLR <13>** to clear Dot Filter Counter A.
- Pulse **Hardware Measurement System U820-31 DFCLKA <13>** (via **Interrupt Latch U1100B-5 MPUSTB <13>**) to load the input latch value of *6hex* into Dot Filter Counter A. This should also set U820-32 DFOUTA high.
- Verify that data bit 1 of **A/D Status Register <13>** is high.
- Write *3FFFhex* to **Comparator Control Voltage DAC U711 <10>** to set **Hardware Measurement System U820-22 COMPA <13>** low.
- Pulse **Hardware Measurement System U820-31 DFCLKA <13>** until it causes the dot filter counter to count up to the value *6hex*, each time verifying that data bit 1 of **A/D Status Register <13>** is still high. (On the first pass through this step, when the counter is at *6hex*, this step is skipped.)
- Pulse **Hardware Measurement System U820-31 DFCLKA <13>** once more (causing the dot filter counter to reach terminal count and clock the low on COMPA through to DFOUTA. This also causes the dot filter counter to be reloaded with the original value).
- Verify that data bit 1 of **A/D Status Register <13>** is now low.
- Write *0000hex* to **Comparator Control Voltage DAC U711 <10>** to set **Hardware Measurement System U820-22 COMPA <13>** high again.
- Pulse **Hardware Measurement System U820-31 DFCLKA <13>** until it causes the dot filter counter to count up to the value *6hex*, each time verifying that data bit 1 of **A/D Status Register <13>** is still low. (On the first pass through this step, when the counter is at *6hex*, this step is skipped.)
- Pulse **Hardware Measurement System U820-31 DFCLKA <13>** once more (causing the dot filter counter to reach terminal count and clock the high on COMPA through to DFOUTA).

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

- Verify that data bit 1 of A/D Status Register <13> is now high.
6. Repeat step 5 six more times replacing the value loaded into the dot filter counter with 5, 4, 3, 2, 1, 0.
 7. Verify that the Channel A Transition Found circuitry of **Hardware Measurement System U820**<13> functions correctly.
 - Set the channel A transition found slope detection in **Hardware Measurement System U820** <13> to positive.
 - Load one of the **Hardware Measurement System LCA U820** <13> input latches with *6hex* (for the dot filter counter) and another with *Ehex* (for the transition found counter).
 - Write *3FFFhex* to **Comparator Control Voltage DAC U711** <10> to set **Hardware Measurement System U820-22 COMPA** <13> low.
 - Pulse **Hardware Measurement System U820-2 DLACLR** <13> to clear Dot Filter Counter A and Transition Found Counter A.
 - Enable the channel A Transition Found interrupt by setting **Enable Register U801-16 TAINTEH** <13> high.
 - Pulse **Hardware Measurement System U820-31 DFCLKA** <13> to load the transition found and dot filter counters with the values from the input latches. This should also clear U820-29 TRFA.
 - Verify that data bit 5 of A/D Status Register <13> is low.
 - Write *0000hex* to **Comparator Control Voltage DAC U711** <10> to set **Hardware Measurement System U820-22 COMPA** <13> high.
 - Pulse **Hardware Measurement System U820-31 DFCLKA** <13> once to cause the transition found counter to count up. If the count is not *Fhex*, verify that data bit 5 of A/D Status Register <13> is low and that no transition A interrupt has occurred (via U931C-6 IRQ). If the count has reached *Fhex*, thus causing the transition found counter to reach terminal count and produce a high at U820-29 TRFA, then

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

verify that data bit 5 of A/D Status Register <13> is high and that a transition A interrupt has occurred.

- Write $3FFF_{hex}$ to Comparator Control Voltage DAC U711 <10> to set Hardware Measurement System U820-22 COMPA <13> low.
- Pulse Hardware Measurement System U820-31 DFCLKA <13> once more.
- If the transition found counter has not yet reached terminal count (i.e. F_{hex}), then verify that data bit 5 of A/D Status Register <13> is low (i.e. that the proper slope is being detected).
- Repeat the previous five items until the transition found counter reaches terminal count (i.e. F_{hex}).
- Set the channel A transition found slope detection in Hardware Measurement System U820 <13> to negative.
- Write 0000_{hex} to Comparator Control Voltage DAC U711 <10> to set Hardware Measurement System U820-22 COMPA <13> high.
- Pulse Hardware Measurement System U820-2 DLACLR <13> to clear Dot Filter Counter A and Transition Found Counter A.
- Pulse Hardware Measurement System U820-31 DFCLKA <13> to load the transition found counter with the value from the input latch (i.e. E_{hex} on the first pass). This should also clear U820-29 TRFA.
- Verify that data bit 5 of A/D Status Register <13> is low.
- Write $3FFF_{hex}$ to Comparator Control Voltage DAC U711 <10> to set Hardware Measurement System U820-22 COMPA <13> low.
- Pulse Hardware Measurement System U820-31 DFCLKA <13> once to cause the transition found counter to count up. If the count is not F_{hex} , verify that data bit 5 of A/D Status Register <13> is low. If the count has reached F_{hex} , thus causing the transition found counter

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

to reach terminal count and produce a high at U820-29 TRFA, then verify that data bit 5 of A/D Status Register <13> is high.

- Write 0000_{hex} to **Comparator Control Voltage DAC U711 <10>** to set **Hardware Measurement System U820-22 COMPA <13>** high.
 - Pulse **Hardware Measurement System U820-31 DFCLKA <13>** once more.
 - If the transition found counter has not yet reached terminal count (i.e. *Fhex*), then verify that data bit 5 of A/D Status Register <13> is low (i.e. that the proper slope is being detected).
 - Repeat the previous five items until the transition found counter reaches terminal count (i.e. *Fhex*).
 - Repeat all the previous items of this step (step 7), replacing the value loaded into the Transition Found Counter A (i.e. *Ehex*) with *Dhex, Chex, ..., 1hex, 0hex*.
 - Read Dot Found Counter A from **Hardware Measurement System U820 <13>**.
 - Pulse **Hardware Measurement System U820-31 DFCLKA <13>** ten times.
 - Read Dot Found Counter A from **Hardware Measurement System U820 <13>** again and verify that this value is the same as the previous value.
9. Verify that the Channel A Dot Found circuitry of **Hardware Measurement System U820<13>** functions correctly.
- Pulse **Hardware Measurement System U820-2 DLACLR <13>** to clear Dot Found Counter A.
 - Read Dot Found Counter A from **Hardware Measurement System U820 <13>** and verify that it is zero.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

- Pulse **Hardware Measurement System** U820-31 DFCLKA <13> once and verify that Dot Found Counter A is 0001_{hex}.
- Pulse **Hardware Measurement System** U820-31 DFCLKA <13> once more (for a total of two) and verify that Dot Found Counter A is now 0002_{hex}.
- Repeat the previous item 14 more times, doubling the number of pulses each time and checking for the appropriate Dot Found Counter A value (i.e. 0004_{hex}, 0008_{hex},..., 4000_{hex}, 8000_{hex}).

Error Index
[m | a-d][1-8]521

The **Comparator Control Voltage** from DAC U711 <10> and the ground reference voltage from **Chan Input Mux A** <9> were not within acceptable range of each other, as indicated by **A Meas Comparator U400** <9>.

Error Index
[m | a-d][1-8]522

The **Comparator Control Voltage** from DAC U711 <10> and the -2V reference voltage from **Chan Input Mux A** <9> were not within acceptable range of each other, as indicated by **A Meas Comparator U400** <9>.

Error Index
[m | a-d][1-8]523

The **Comparator Control Voltage** from DAC U711 <10> and the +2V reference voltage from **Chan Input Mux A** <9> were not within acceptable range of each other, as indicated by **A Meas Comparator U400** <9>.

Error Index
[m | a-d][1-8]524

Dot Filter A of **Hardware Measurement System** U820 <13> did not load correctly.

Error Index
[m | a-d][1-8]525

Dot Filter A of **Hardware Measurement System** U820 <13> did not correctly count events.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter A (m[1-2]52X)
Acq[1-8]	Measurement	Filter A (a[1-8]52X)
Acq[1-8]	Measurement	Filter A (b[1-8]52X)
Acq[1-8]	Measurement	Filter A (c[1-8]52X)
Acq[1-8]	Measurement	Filter A (d[1-8]52X)

Error Index [m a-d][1-8]526	Transition Found Counter A of Hardware Measurement System U820 <13> did not load correctly.
----------------------------------	--

Error Index [m a-d][1-8]527	Transition Found Counter A of Hardware Measurement System U820 <13> did not count transitions correctly.
----------------------------------	---

Error Index [m a-d][1-8]528	The Acquisition processor never saw an interrupt from Transition Found Counter A of Hardware Measurement System U820 <13> when it should have; or the processor saw an interrupt when it shouldn't have.
----------------------------------	---

Error Index [m a-d][1-8]529	Transition Found Counter A of Hardware Measurement System U820 <13> did not count transitions on the correct slope.
----------------------------------	--

Error Index [m a-d][1-8]52A	Dot Found Counter A of Hardware Measurement System U820 <13> could not be cleared.
----------------------------------	---

Error Index [m a-d][1-8]52B	Dot Found Counter A of Hardware Measurement System U820 <13> did not count correctly.
----------------------------------	--

Error Index [m a-d][1-8]52C	Dot Found Counter A of Hardware Measurement System U820 <13> did not stop counting when a "transition found" occurred.
----------------------------------	---

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

Routine Name Filter B

Overview This test verifies **B Meas Comparator <9>** (by using three voltage reference inputs) and the Dot Filter, Transition Found, and Dot Found capabilities of **Hardware Measurement System LCA U820 <13>** channel B. All of this is accomplished by configuring the channel B measurement system, simulating hardware strobes, and checking the results read back through **A/D Status Register <13>**.

Description

1. Configure hardware measurement channel B.
 - Set **Measurement & A/D Strobe Generator U901 <13>** pins 16 & 17 low to disable **Chan A/D A** and **Chan A/D B <9>**.
 - Set **Measurement & A/D Strobe Generator U901 <13>** pins 14 & 15 low to disable the **Hardware Measurement System LCA U820** Dot Filters (via disabling U820-31 DFCLKA and U820-37 DFCLKB).
 - Set **Measurement & A/D Strobe Generator U901 <13>** pins 12 & 13 high.
 - Set **Measurement & A/D Strobe Generator U901-18 <13>** low to disable **A Meas Comparator <9>**.
 - Set **Measurement & A/D Strobe Generator U901-19 <13>** high to enable **B Meas Comparator <9>**.
2. Verify that **B Meas Comparator <9>** correctly sees the ground voltage reference.
 - Enable the processor-generated strobe (MPUSTB) by setting **Enable Register U801-18 MSTREN(L) <13>** low.
 - Set **Chan Mux Latch U1501 <9>** pins 17, 18, and 19 low to select the ground voltage reference at **Chan Input Mux B U100-1 <9>**.
 - Write **1E7A_{hex}** to **Comparator Control Voltage DAC U710.<10>** and wait a short amount of time for the DAC to settle.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

- Pulse B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U200.
 - Verify that data bit 2 (COMPB) of A/D Status Register <13> is low.
 - Write 2186_{hex} to Comparator Control Voltage DAC U710.<10> and wait a short amount of time for the DAC to settle.
 - Pulse B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U200.
 - Verify that data bit 2 (COMPB) of A/D Status Register <13> is high.
3. Verify that B Meas Comparator <9> correctly sees the -2 volt reference.
- Set Chan Mux Latch U1501 <9> pins 17 & 19 low and pin 18 high to select -2V REF at Chan Input Mux B U100-3 <9>.
 - Write 0000_{hex} to Comparator Control Voltage DAC U710.<10> and wait a short amount of time for the DAC to settle.
 - Pulse B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U200.
 - Verify that data bit 2 (COMPB) of A/D Status Register <13> is low.
 - Write 030C_{hex} to Comparator Control Voltage DAC U710.<10> and wait a short amount of time for the DAC to settle.
 - Pulse B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U200.
 - Verify that data bit 2 (COMPB) of A/D Status Register <13> is high.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

4. Verify that B Meas Comparator <9> correctly sees the +2 volt reference.
 - Set Chan Mux Latch U1501 <9> pin 17 low and pins 18 & 19 high to select +2V REF at Chan Input Mux B U100-4 <9>.
 - Write 3FFF_{hex} to Comparator Control Voltage DAC U710.<10> and wait a short amount of time for the DAC to settle.
 - Pulse B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U200.
 - Verify that data bit 2 (COMPB) of A/D Status Register <13> is high.
 - Write 3CF4_{hex} to Comparator Control Voltage DAC U710.<10> and wait a short amount of time for the DAC to settle.
 - Pulse B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) to validate the output of U200.
 - Verify that data bit 2 (COMPB) of A/D Status Register <13> is low.
5. Verify that the Channel B Dot Filter of Hardware Measurement System U820 <13> functions correctly.
 - Set Measurement & A/D Strobe Generator U901-15 <13> high to enable the Dot Filter Clock B (DFCLKB).
 - Set Chan Mux Latch U1501 <9> pins 17, 18, and 19 low to select the ground voltage reference at Chan Input Mux B U100-1 <9>.
 - Load one of the Hardware Measurement System LCA U820 <13> input latches with 6_{hex} (for the dot filter counter).
 - Write 0000_{hex} to Comparator Control Voltage DAC U710 <10> to set Hardware Measurement System U820-24 COMPB <13> high (via B Meas Comparator U200-7 COMPB <9>).

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

- Pulse Hardware Measurement System U820-63 DLBCLR <13> to clear Dot Filter Counter B.
- Pulse Hardware Measurement System U820-37 DFCLKB <13> (via Interrupt Latch U1100B-5 MPUSTB <13>) to load the input latch value of *6hex* into Dot Filter Counter B. This should also set U820-36 DFOUTB high.
- Verify that data bit 3 of A/D Status Register <13> is high.
- Write *3FFFhex* to Comparator Control Voltage DAC U710.<10> to set Hardware Measurement System U820-24 COMPB <13> low.
- Pulse Hardware Measurement System U820-37 DFCLKB <13> until it causes the dot filter counter to count up to the value *6hex*, each time verifying that data bit 3 of A/D Status Register <13> is still high. (On the first pass through this step, when the counter is at *6hex*, this step is skipped.)
- Pulse Hardware Measurement System U820-37 DFCLKB <13> once more (causing the dot filter counter to reach terminal count and clock the low on COMPB through to DFOUTB. This also causes the dot filter counter to be reloaded with the original value).
- Verify that data bit 3 of A/D Status Register <13> is now low.
- Write *0000hex* to Comparator Control Voltage DAC U710 <10> to set Hardware Measurement System U820-24 COMPB <13> high again.
- Pulse Hardware Measurement System U820-37 DFCLKB <13> until it causes the dot filter counter to count up to the value *6hex*, each time verifying that data bit 3 of A/D Status Register <13> is still low. (On the first pass through this step, when the counter is at *6hex*, this step is skipped.)
- Pulse Hardware Measurement System U820-37 DFCLKB <13> once more (causing the dot filter counter to reach terminal count and clock the high on COMPB through to DFOUTB).

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

- Verify that data bit 3 of A/D Status Register <13> is now high.
6. Repeat step 5 six more times replacing the value loaded into the dot filter counter with 5, 4, 3, 2, 1, 0.
 7. Verify that the Channel B Transition Found circuitry of **Hardware Measurement System U820**<13> functions correctly.
 - Set the channel B transition found slope detection in **Hardware Measurement System U820** <13> to positive.
 - Load one of the **Hardware Measurement System LCA U820** <13> input latches with *6hex* (for the dot filter counter) and another with *Ehex* (for the transition found counter).
 - Write *3FFFhex* to **Comparator Control Voltage DAC U710**.<10> to set **Hardware Measurement System U820-24 COMPB** <13> low.
 - Pulse **Hardware Measurement System U820-63 DLBCLR** <13> to clear Dot Filter Counter B and Transition Found Counter B.
 - Enable the channel B Transition Found interrupt by setting **Enable Register U801-16 TBINTEN(H)** <13> high.
 - Pulse **Hardware Measurement System U820-37 DFCLKB** <13> to load the transition found and dot filter counters with the values from the input latches. This should also clear U820-39 TRFB.
 - Verify that data bit 4 of A/D Status Register <13> is low.
 - Write *0000hex* to **Comparator Control Voltage DAC U710**.<10> to set **Hardware Measurement System U820-24 COMPB** <13> high.
 - Pulse **Hardware Measurement System U820-37 DFCLKB** <13> once to cause the transition found counter to count up. If the count is not *Fhex*, verify that data bit 4 of A/D Status Register <13> is low and that no transition B interrupt has occurred (via U931D-8 IRQ). If the count has reached *Fhex*, thus causing the transition found counter to reach terminal count and produce a high at U820-39 TRFB, then

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

verify that data bit 4 of A/D Status Register <13> is high and that a transition B interrupt has occurred.

- Write 3FFF_{hex} to Comparator Control Voltage DAC U710.<10> to set Hardware Measurement System U820-24 COMPB <13> low.
- Pulse Hardware Measurement System U820-37 DFCLKB <13> once more.
- If the transition found counter has not yet reached terminal count (i.e. F_{hex}), then verify that data bit 4 of A/D Status Register <13> is low (i.e. that the proper slope is being detected).
- Repeat the previous five items until the transition found counter reaches terminal count (i.e. F_{hex}).
- Set the channel B transition found slope detection in Hardware Measurement System U820 <13> to negative.
- Write 0000_{hex} to Comparator Control Voltage DAC U710.<10> to set Hardware Measurement System U820-24 COMPB <13> high.
- Pulse Hardware Measurement System U820-63 DLBCLR <13> to clear Dot Filter Counter B and Transition Found Counter B.
- Pulse Hardware Measurement System U820-37 DFCLKB <13> to load the transition found counter with the value from the input latch (i.e. E_{hex} on the first pass). This should also clear U820-39 TRFB.
- Verify that data bit 4 of A/D Status Register <13> is low.
- Write 3FFF_{hex} to Comparator Control Voltage DAC U710.<10> to set Hardware Measurement System U820-24 COMPB <13> low.
- Pulse Hardware Measurement System U820-37 DFCLKB <13> once to cause the transition found counter to count up. If the count is not F_{hex}, verify that data bit 4 of A/D Status Register <13> is low. If the count has reached F_{hex}, thus causing the transition found counter

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

to reach terminal count and produce a high at U820-39 TRFB, then verify that data bit 4 of A/D Status Register <13> is high.

- Write 0000 $_{hex}$ to **Comparator Control Voltage DAC** U710.<10> to set **Hardware Measurement System** U820-24 COMPB <13> high.
 - Pulse **Hardware Measurement System** U820-37 DFCLKB <13> once more.
 - If the transition found counter has not yet reached terminal count (i.e. F_{hex}), then verify that data bit 4 of A/D Status Register <13> is low (i.e. that the proper slope is being detected).
 - Repeat the previous five items until the transition found counter reaches terminal count (i.e. F_{hex}).
 - Repeat all the previous items of this step (step 7), replacing the value loaded into the Transition Found Counter B (i.e. E_{hex}) with D_{hex} , C_{hex} , ..., 1_{hex} , 0_{hex} .
 - Read Dot Found Counter B from **Hardware Measurement System** U820 <13>.
 - Pulse **Hardware Measurement System** U820-37 DFCLKB <13> ten times.
 - Read Dot Found Counter B from **Hardware Measurement System** U820 <13> again and verify that this value is the same as the previous value.
9. Verify that the Channel B Dot Found circuitry of **Hardware Measurement System** U820<13> functions correctly.
- Pulse **Hardware Measurement System** U820-63 DLBCLR <13> to clear Dot Found Counter B.
 - Read Dot Found Counter B from **Hardware Measurement System** U820 <13> and verify that it is zero.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

- Pulse **Hardware Measurement System** U820-37 DFCLKB <13> once and verify that Dot Found Counter B is 0001_{hex}.
- Pulse **Hardware Measurement System** U820-37 DFCLKB <13> once more (for a total of two) and verify that Dot Found Counter B is now 0002_{hex}.
- Repeat the previous item 14 more times, doubling the number of pulses each time and checking for the appropriate Dot Found Counter B value (i.e. 0004_{hex}, 0008_{hex},..., 4000_{hex}, 8000_{hex}).

Error Index
[m | a-d][1-8]531

The **Comparator Control Voltage** from DAC U710 <10> and the ground reference voltage from **Chan Input Mux B** <9> were not within acceptable range of each other, as indicated by **B Meas Comparator U200** <9>.

Error Index
[m | a-d][1-8]532

The **Comparator Control Voltage** from DAC U710 <10> and the -2V reference voltage from **Chan Input Mux B** <9> were not within acceptable range of each other, as indicated by **B Meas Comparator U200** <9>.

Error Index
[m | a-d][1-8]533

The **Comparator Control Voltage** from DAC U710 <10> and the +2V reference voltage from **Chan Input Mux B** <9> were not within acceptable range of each other, as indicated by **B Meas Comparator U200** <9>.

Error Index
[m | a-d][1-8]534

Dot Filter B of **Hardware Measurement System** U820 <13> did not load correctly.

Error Index
[m | a-d][1-8]535

Dot Filter B of **Hardware Measurement System** U820 <13> did not correctly count events.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	Filter B (m[1-2]53X)
Acq[1-8]	Measurement	Filter B (a[1-8]53X)
Acq[1-8]	Measurement	Filter B (b[1-8]53X)
Acq[1-8]	Measurement	Filter B (c[1-8]53X)
Acq[1-8]	Measurement	Filter B (d[1-8]53X)

Error Index [m a-d][1-8]536	Transition Found Counter B of Hardware Measurement System U820 <13> did not load correctly.
----------------------------------	---

Error Index [m a-d][1-8]537	Transition Found Counter B of Hardware Measurement System U820 <13> did not count transitions correctly.
----------------------------------	--

Error Index [m a-d][1-8]538	The Acquisition processor never saw an interrupt from Transition Found Counter B of Hardware Measurement System U820 <13> when it should have; or the processor saw an interrupt when it shouldn't have.
----------------------------------	--

Error Index [m a-d][1-8]539	Transition Found Counter B of Hardware Measurement System U820 <13> did not count transitions on the correct slope.
----------------------------------	---

Error Index [m a-d][1-8]53A	Dot Found Counter B of Hardware Measurement System U820 <13> could not be cleared.
----------------------------------	--

Error Index [m a-d][1-8]53B	Dot Found Counter B of Hardware Measurement System U820 <13> did not count correctly.
----------------------------------	---

Error Index [m a-d][1-8]53C	Dot Found Counter B of Hardware Measurement System U820 <13> did not stop counting when a "transition found" occurred.
----------------------------------	--

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	A->B B->A (m[1-2]54X)
Acq[1-8]	Measurement	A->B B->A (a[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (b[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (c[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (d[1-8]54X)

Routine Name A->B B->A

Overview This test verifies that the hardware logic in **Measurement & A/D Strobe Generator** <13> necessary for B after A and A after B transition counting functions correctly.

Description

1. Configure hardware measurement channels A and B.
 - Set **Measurement & A/D Strobe Generator** U901 <13> pins 16 & 17 low to disable Chan A/D A and Chan A/D B <9>.
 - Set **Measurement & A/D Strobe Generator** U901 <13> pins 14 & 15 high to enable the Dot Filters in **Hardware Measurement System** LCA U820 (via enabling U820-31 DFCLKA and U820-37 DFCLKB).
 - Set **Measurement & A/D Strobe Generator** U901 <13> pins 12 & 13 high to disable A after B and B after A transitions.
 - Set **Measurement & A/D Strobe Generator** U901 <13> pins 18 & 19 high to enable A Meas Comparator and B Meas Comparator <9>.
 - Set **Chan Mux Latch** U1501 <9> pins 14, 15, 16, 17, 18, and 19 low to select the ground voltage references at Chan Input Mux A U301-1 <9> and Chan Input Mux B U100-1 <9>.
 - Load one of the **Hardware Measurement System** LCA U820 <13> input latches with *6hex* (for each dot filter counter) and another with *Ehex* (for each transition found counter).
2. Check B after A transition counting logic.
 - Write *3FFFhex* to **Comparator Control Voltage** DACs U710 and U711 <10> to set **Hardware Measurement System** U820-22 COMPA and U820-24 COMPB <13> low.
 - Pulse **Hardware Measurement System** U820-2 DLACLR and U820-63 DLBCLR <13> to clear the dot filter and transition found counters.
 - Pulse **Hardware Measurement System** U820-31 DFCLKA and U820-37 DFCLKB <13> to load the dot filter and transition found

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	A->B B->A (m[1-2]54X)
Acq[1-8]	Measurement	A->B B->A (a[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (b[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (c[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (d[1-8]54X)

counters with the values from the input latches. This should also clear U820-29 TRFA and U820-39 TRFB.

- Set **Hardware Measurement System** U901-13 <13> low to enable B after A transition counting.
 - Verify that **A/D Status Register** U800-13 (TRFA) is low and U800-11 (TRFB) is low (i.e. that no transitions have occurred).
 - Write 0000_{hex} to **Comparator Control Voltage** DACs U710 and U711 <10> to set **Hardware Measurement System** U820-22 COMPA and U820-24 COMPB <13> high.
 - Pulse **Hardware Measurement System** U820-31 DFCLKA and U820-37 DFCLKB <13> once.
 - Verify that **A/D Status Register** U800-13 (TRFA) is high and U800-11 (TRFB) is low (i.e. that a transition occurred on channel A only).
 - Write 3FFF_{hex} to **Comparator Control Voltage** DACs U710 and U711 <10> to set **Hardware Measurement System** U820-22 COMPA and U820-24 COMPB <13> low again.
 - Pulse **Hardware Measurement System** U820-31 DFCLKA and U820-37 DFCLKB <13> once.
 - Write 0000_{hex} to **Comparator Control Voltage** DACs U710 and U711 <10> to set **Hardware Measurement System** U820-22 COMPA and U820-24 COMPB <13> high again.
 - Pulse **Hardware Measurement System** U820-31 DFCLKA and U820-37 DFCLKB <13> once.
 - Verify that **A/D Status Register** U800-13 (TRFA) is high and U800-11 (TRFB) is high (i.e. that transitions have occurred on A and B).
 - Set **Hardware Measurement System** U901-13 <13> high to disable B after A transition counting.
3. Check A after B transition counting logic.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Measurement	A->B B->A (m[1-2]54X)
Acq[1-8]	Measurement	A->B B->A (a[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (b[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (c[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (d[1-8]54X)

- Write `3FFFhex` to Comparator Control Voltage DACs U710 and U711 <10> to set Hardware Measurement System U820-22 COMPA and U820-24 COMPB <13> low.
- Pulse Hardware Measurement System U820-2 DLACLR and U820-63 DLBCLR <13> to clear the dot filter and transition found counters.
- Pulse Hardware Measurement System U820-31 DFCLKA and U820-37 DFCLKB <13> to load the dot filter and transition found counters with the values from the input latches. This should also clear U820-29 TRFA and U820-39 TRFB.
- Set Hardware Measurement System U901-12 <13> low to enable A after B transition counting.
- Verify that A/D Status Register U800-13 (TRFA) is low and U800-11 (TRFB) is low (i.e. that no transitions have occurred).
- Write `0000hex` to Comparator Control Voltage DACs U710 and U711 <10> to set Hardware Measurement System U820-22 COMPA and U820-24 COMPB <13> high.
- Pulse Hardware Measurement System U820-31 DFCLKA and U820-37 DFCLKB <13> once.
- Verify that A/D Status Register U800-13 (TRFA) is low and U800-11 (TRFB) is high (i.e. that a transition occurred on channel B only).
- Write `3FFFhex` to Comparator Control Voltage DACs U710 and U711 <10> to set Hardware Measurement System U820-22 COMPA and U820-24 COMPB <13> low again.
- Pulse Hardware Measurement System U820-31 DFCLKA and U820-37 DFCLKB <13> once.
- Write `0000hex` to Comparator Control Voltage DACs U710 and U711 <10> to set Hardware Measurement System U820-22 COMPA and U820-24 COMPB <13> high again.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Measurement	A->B B->A (m[1-2]54X)
Acq[1-8]	Measurement	A->B B->A (a[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (b[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (c[1-8]54X)
Acq[1-8]	Measurement	A->B B->A (d[1-8]54X)

- Pulse Hardware Measurement System U820-31 DFCLKA and U820-37 DFCLKB <13> once.
- Verify that A/D Status Register U800-13 (TRFA) is high and U800-11 (TRFB) is high (i.e. that transitions have occurred on A and B).

Error Index
[m | a-d][1-8]541

A transition event occurred on channel A or B when it shouldn't have (just after the measurement channels were configured for B after A), or a transition event did not occur on channel A when it should have.

Error Index
[m | a-d][1-8]542

A transition event did not occur on channel B (when the measurement channels were configured for B after A) when it should have (i.e. after the transition on channel A).

Error Index
[m | a-d][1-8]543

A transition event occurred on channel A or B when it shouldn't have (just after the measurement channels were configured for A after B), or a transition event did not occur on channel B when it should have.

Error Index
[m | a-d][1-8]544

A transition event did not occur on channel A (when the measurement channels were configured for A after B) when it should have (i.e. after the transition on channel B).

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 1 (m[1-2]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (a[1-8]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (b[1-8]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (c[1-8]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (d[1-8]61X)

Routine Name Cntrl Func 1 (Control Function 1)

Overview This test verifies that **Sampling Head Blow By Control Voltage** <10> functions correctly by using a "walking ones" type test on each DAC of U1101.

Description

1. Write 08_{hex} to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1220-4 SHBB1/5 <11> as the input to **Diagnostic A/D** <11>.
2. Write the test pattern 01_{hex} to **Sampling Head Blow By Control Voltage** U1101 part a <10>. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing **Diagnostic A/D** U1410 <11>.
4. Read the ADC code from **Diagnostic A/D** U1410 <11>.
 - Wait until **A/D Status Register** U1300 <11> pin 2 is high.
 - Read **Diagnostic A/D** through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns 02_{hex}, 04_{hex}, 08_{hex}, 10_{hex}, 20_{hex}, 40_{hex}, 80_{hex}.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write 09_{hex} to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1220-5 SHBB2/6 <11> as the input to **Diagnostic A/D** <11>.
8. Repeat steps 2 through 6 using **Sampling Head Blow By Control Voltage** U1101 part b <10>.
9. Write 0A_{hex} to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1220-6 SHBB3/7 <11> as the input to **Diagnostic A/D** <11>.
10. Repeat steps 2 through 6 using **Sampling Head Blow By Control Voltage** U1101 part c <10>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 1 (m[1-2]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (a[1-8]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (b[1-8]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (c[1-8]61X)
Acq[1-8]	Sample Head	Cntrl Func 1 (d[1-8]61X)

11. Write 0B_{hex} to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1220-7 SHBB4/8 <11> as the input to Diagnostic A/D <11>.
12. Repeat steps 2 through 6 using Sampling Head Blow By Control Voltage U1101 part d <10>.

Error Index

[m | a-d][1-8]611

The Sampling Head Blow By Control Voltage U1101 part a <10> did not function as expected.

Error Index

[m | a-d][1-8]612

The Sampling Head Blow By Control Voltage U1101 part b <10> did not function as expected.

Error Index

[m | a-d][1-8]613

The Sampling Head Blow By Control Voltage U1101 part c <10> did not function as expected.

Error Index

[m | a-d][1-8]614

The Sampling Head Blow By Control Voltage U1101 part d <10> did not function as expected.

Sampling Head Capabilities vs. Control Function 1

	part a (Head 0 ACV8)	part b (Head 0 ACV9)	part c (Head 1 ACV8)	part d (Head 1 ACV9)
Blowby	Ch 1 Blowby	Ch 2 Blowby	Ch 1 Blowby	Ch 2 Blowby
Pulse Gen	Ch 1 Risetime	Ch 2 Risetime	Ch 1 Risetime	Ch 2 Risetime

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 2 (m[1-2]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (a[1-8]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (b[1-8]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (c[1-8]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (d[1-8]62X)

Routine Name Cntrl Func 2 (Control Function 2)

Overview This test verifies that **Sampling Head Loop Gain Control Voltage** <10> functions correctly by using a "walking ones" type test on each DAC of U1100.

Description

1. Write *0Chex* to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1220-12 SHLG1/5 <11> as the input to **Diagnostic A/D** <11>.
2. Write the test pattern *01hex* to **Sampling Head Blow By Control Voltage** U1101 part a <10>. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing **Diagnostic A/D** U1410 <11>.
4. Read the ADC code from **Diagnostic A/D** U1410 <11>.
 - Wait until **A/D Status Register** U1300 <11> pin 2 is high.
 - Read **Diagnostic A/D** through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns *02hex*, *04hex*, *08hex*, *10hex*, *20hex*, *40hex*, *80hex*.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write *0Dhex* to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1220-11 SHLG2/6 <11> as the input to **Diagnostic A/D** <11>.
8. Repeat steps 2 through 6 using **Sampling Head Loop Gain Control Voltage** U1100 part b <10>.
9. Write *0Ehex* to **Diagnostic Mux Control Latch** U1401 <11> to select **Diagnostic Input Mux** U1220-10 SHLG3/7 <11> as the input to **Diagnostic A/D** <11>.
10. Repeat steps 2 through 6 using **Sampling Head Loop Gain Control Voltage** U1100 part c <10>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 2 (m[1-2]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (a[1-8]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (b[1-8]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (c[1-8]62X)
Acq[1-8]	Sample Head	Cntrl Func 2 (d[1-8]62X)

11. Write 0F_{hex} to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1220-9 SHLG4/8 <11> as the input to Diagnostic A/D <11>.
12. Repeat steps 2 through 6 using Sampling Head Loop Gain Control Voltage U1100 part d <10>.

Error Index
[m | a-d][1-8]621

The Sampling Head Loop Gain Control Voltage U1100 part a <10> did not function as expected.

Error Index
[m | a-d][1-8]622

The Sampling Head Loop Gain Control Voltage U1100 part b <10> did not function as expected.

Error Index
[m | a-d][1-8]623

The Sampling Head Loop Gain Control Voltage U1100 part c <10> did not function as expected.

Error Index
[m | a-d][1-8]624

The Sampling Head Loop Gain Control Voltage U1100 part d <10> did not function as expected.

Sampling Head Capabilities vs. Control Function 2

	part a (Head 0 ACV7)	part b (Head 0 ACV6)	part c (Head 1 ACV7)	part d (Head 1 ACV6)
Acquisition	Ch 1 Loop Gain	Ch 2 Loop Gain	Ch 1 Loop Gain	Ch 2 Loop Gain
Pulse Gen	Ch 1 Falltime	Ch 2 Falltime	Ch 1 Falltime	Ch 2 Falltime

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 3 (m[1-2]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (a[1-8]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (b[1-8]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (c[1-8]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (d[1-8]63X)

Routine Name Cntrl Func 3 (Control Function 3)

Overview This test verifies that **Sampling Head Offset Control Voltage** <10> functions correctly by using a "walking ones" type test on each DAC of U901 and U900.

Description

1. Write 04_{hex} to **Diagnostic Mux Control Latch U1401** <11> to select **Diagnostic Input Mux U1120-12 SHOFF1/5** <11> as the input to **Diagnostic A/D** <11>.
2. Write the test pattern 010_{hex} to **Sampling Head Offset Control Voltage U901** part a <10>. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing **Diagnostic A/D U1410** <11>.
4. Read the ADC code from **Diagnostic A/D U1410** <11>.
 - Wait until **A/D Status Register U1300** <11> pin 2 is high.
 - Read **Diagnostic A/D** through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns 020_{hex}, 040_{hex}, 080_{hex}, 100_{hex}, 200_{hex}, 400_{hex}, 800_{hex}.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write 07_{hex} to **Diagnostic Mux Control Latch U1401** <11> to select **Diagnostic Input Mux U1120-9 SHOFF2/6** <11> as the input to **Diagnostic A/D** <11>.
8. Repeat steps 2 through 6 using **Sampling Head Offset Control Voltage U901** part b <10>.
9. Write 05_{hex} to **Diagnostic Mux Control Latch U1401** <11> to select **Diagnostic Input Mux U1120-11 SHOFF3/7** <11> as the input to **Diagnostic A/D** <11>.
10. Repeat steps 2 through 6 using **Sampling Head Offset Control Voltage U900** part a <10>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 3 (m[1-2]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (a[1-8]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (b[1-8]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (c[1-8]63X)
Acq[1-8]	Sample Head	Cntrl Func 3 (d[1-8]63X)

11. Write 06hex to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1120-10 SHOFF4/8 <11> as the input to Diagnostic A/D <11>.
12. Repeat steps 2 through 6 using Sampling Head Offset Control Voltage U900 part b <10>.

Error Index
[m | a-d][1-8]631

The Sampling Head Offset Control Voltage U901 part a <10> did not function as expected.

Error Index
[m | a-d][1-8]632

The Sampling Head Offset Control Voltage U901 part b <10> did not function as expected.

Error Index
[m | a-d][1-8]633

The Sampling Head Offset Control Voltage U900 part a <10> did not function as expected.

Error Index
[m | a-d][1-8]634

The Sampling Head Offset Control Voltage U900 part b <10> did not function as expected.

Sampling Head Capabilities vs. Control Function 3

	U901 part a (Head 0 ACV1)	U901 part b (Head 0 ACV5)	U900 part a (Head 1 ACV1)	U900 part b (Head 1 ACV5)
Acquisition	Ch 1 Offset Null	Ch 2 Offset Null	Ch 1 Offset Null	Ch 2 Offset Null
Pulse Gen	Ch 1 Offset	Ch 2 Offset	Ch 1 Offset	Ch 2 Offset
Trigger Sync	Ch 1 Sync	Ch 2 Sync	Ch 1 Sync	Ch 2 Sync
Trigger Level	Ch 1 Level	Ch 2 Level	Ch 1 Level	Ch 2 Level

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 4 (m[1-2]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (a[1-8]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (b[1-8]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (c[1-8]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (d[1-8]64X)

Routine Name Cntrl Func 4 (Control Function 4)

Overview This test verifies that Sampling Head TDR Level Control Voltage <10> functions correctly by using a "walking ones" type test on each DAC of U1000.

Description

1. Write 00*hex* to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1120-4 SHTDR1/5 <11> as the input to Diagnostic A/D <11>.
2. Write the test pattern 01*hex* to Sampling Head TDR Level Control Voltage U1101 part a <10>. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing Diagnostic A/D U1410 <11>.
4. Read the ADC code from Diagnostic A/D U1410 <11>.
 - Wait until A/D Status Register U1300 <11> pin 2 is high.
 - Read Diagnostic A/D through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns 02*hex*, 04*hex*, 08*hex*, 10*hex*, 20*hex*, 40*hex*, 80*hex*.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write 01*hex* to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1120-5 SHTDR2/6 <11> as the input to Diagnostic A/D <11>.
8. Repeat steps 2 through 6 using Sampling Head TDR Level Control Voltage U1000 part b <10>.
9. Write 02*hex* to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1120-6 SHTDR3/7 <11> as the input to Diagnostic A/D <11>.
10. Repeat steps 2 through 6 using Sampling Head TDR Level Control Voltage U1000 part c <10>.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 4 (m[1-2]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (a[1-8]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (b[1-8]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (c[1-8]64X)
Acq[1-8]	Sample Head	Cntrl Func 4 (d[1-8]64X)

11. Write 03_{hex} to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1120-7 SHTDR4/8 <11> as the input to Diagnostic A/D <11>.
12. Repeat steps 2 through 6 using Sampling Head TDR Level Control Voltage U1000 part d <10>.

Error Index
[m | a-d][1-8]641

The Sampling Head TDR Level Control Voltage U1000 part a <10> did not function as expected.

Error Index
[m | a-d][1-8]642

The Sampling Head TDR Level Control Voltage U1000 part b <10> did not function as expected.

Error Index
[m | a-d][1-8]643

The Sampling Head TDR Level Control Voltage U1000 part c <10> did not function as expected.

Error Index
[m | a-d][1-8]644

The Sampling Head TDR Level Control Voltage U1000 part d <10> did not function as expected.

Sampling Head Capabilities vs. Control Function 4

	part a (Head 0 ACV2)	part b (Head 0 ACV3)	part c (Head 1 ACV2)	part d (Head 1 ACV3)
TDR	Ch 1 TDR Level	Ch 2 TDR Level	Ch 1 TDR Level	Ch 2 TDR Level
Offset Termination	Ch 1 Termination	Ch 2 Termination	Ch 1 Termination	Ch 2 Termination
Pulse Gen	Ch 1 Amplitude	Ch 2 Amplitude	Ch 1 Amplitude	Ch 2 Amplitude

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 5 (m[1-2]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (a[1-8]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (b[1-8]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (c[1-8]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (d[1-8]65X)

Routine Name Cntrl Func 5 (Control Function 5)

Overview This test verifies that Offset Correction & TDR Delay Control Voltage <10> functions correctly by using a "walking ones" type test on part c & d of DAC U1200.

Description

1. Write 13_{hex} to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1330-7 TDRDEL1,2/5,6 <11> as the input to Diagnostic A/D <11>.
2. Write the test pattern 01_{hex} to Offset Correction & TDR Delay Control Voltage U1200 part c <10>. Wait a short amount of time for the DAC to settle.
3. Start the analog-to-digital conversion by reading and writing Diagnostic A/D U1410 <11>.
4. Read the ADC code from Diagnostic A/D U1410 <11>.
 - Wait until A/D Status Register U1300 <11> pin 2 is high.
 - Read Diagnostic A/D through buffer U1400.
5. Repeat steps 2, 3, and 4 using the patterns 02_{hex}, 04_{hex}, 08_{hex}, 10_{hex}, 20_{hex}, 40_{hex}, 80_{hex}.
6. Verify that the ADC codes read in step 4 are within acceptable limits.
7. Write 12_{hex} to Diagnostic Mux Control Latch U1401 <11> to select Diagnostic Input Mux U1330-6 TDRDEL3,4/7,8 <11> as the input to Diagnostic A/D <11>.
8. Repeat steps 2 through 6 using Offset Correction & TDR Delay Control Voltage part d <10>.

Error Index
[m | a-d][1-8]651

The Offset Correction & TDR Delay Control Voltage U1200 part c <10> did not function as expected.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	Cntrl Func 5 (m[1-2]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (a[1-8]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (b[1-8]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (c[1-8]65X)
Acq[1-8]	Sample Head	Cntrl Func 5 (d[1-8]65X)

Error Index

[m | a-d][1-8]652

The Offset Correction & TDR Delay Control Voltage U1200 part d <10> did not function as expected.

Sampling Head Capabilities vs. Control Function 5

	part c (Head 0 ACV4)	part d (Head 1 ACV4)
TDR Delta Delay	TDR Delay	TDR Delay
Pulse Gen Delta Delay	Relative Delay	Relative Delay

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	EEPROM (m[1-2]66X)
Acq[1-8]	Sample Head	EEPROM (a[1-8]66X)
Acq[1-8]	Sample Head	EEPROM (b[1-8]66X)
Acq[1-8]	Sample Head	EEPROM (c[1-8]66X)
Acq[1-8]	Sample Head	EEPROM (d[1-8]66X)

Routine Name EEPROM

Overview This test verifies that the contents of the EEPROM in each head is valid by performing a checksum on their contents.

Description

- Instruct the EEPROM in head 0 to output the first word of memory by serially shifting the instruction 180_{hex} into the EEPROM through the Sampling Head EEPROM Interface <12>.
 - Set U301-5 CLK1,2/5,6 <12> low.
 - Set U301-4 CS1,2/5,6 <12> high to select the EEPROM.
 - Set U301-7 <12> low to enable data driver U300B.
 - Set U301-6 (DI/O1,2/5,6) <12> low.
 - Pulse U301-5 CLK1,2/5,6 <12> to clock a dummy data bit into the EEPROM.
 - Output the MSB of the instruction (180_{hex}) on U301-6 (DI/O1,2/5,6) <12>. Afterwards shift the instruction one bit to the left.
 - Pulse U301-5 CLK1,2/5,6 <12> to clock the instruction bit into the EEPROM.
 - Repeat the previous two steps until all bits of the instruction have been clocked into the EEPROM.
 - Set U301-7 <12> high to disable data driver U300B.
- Read the contents of the EEPROM word by serial shifting it into the processor.
 - Pulse U301-5 CLK1,2/5,6 <12> to read a dummy data bit.
 - Shift the data bit received one bit to the left and add on the next 16 incoming data bits by repeating the previous step, and then this step, 16 more times.
 - Set U301-4 CS1,2/5,6 <12> low to deselect the EEPROM.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Sample Head	EEPROM (m[1-2]66X)
Acq[1-8]	Sample Head	EEPROM (a[1-8]66X)
Acq[1-8]	Sample Head	EEPROM (b[1-8]66X)
Acq[1-8]	Sample Head	EEPROM (c[1-8]66X)
Acq[1-8]	Sample Head	EEPROM (d[1-8]66X)

3. Repeat steps 1 and 2 63 more times, incrementing the instruction pattern each time in order to read all of the EEPROM contents.
4. Repeat steps 1 through 3 to read the contents of head 1, using the appropriate clock, select, data driver, and data lines for head 1 in **Sampling Head EEPROM Interface** (i.e. CS3,4/7,8, etc).
5. If the first word of both EEPROMs is $FFFF_{hex}$, then set the error index code to "not present" and terminate the test.
6. Sum together the first 63 words from head 0. Sum together the first 63 words from head 1. Complement each sum and then add 1 to each sum.
7. Verify that the sum for head 0 matches the contents of the 64th word of head 0. If these two do not match, then terminate the test.
8. Verify that the sum for head 1 matches the contents of the 64th word of head 1.

Error Index

[m | a-d][1-8]661

The checksum for the contents of the EEPROM in head 0 did not match the checksum stored in the EEPROM.

Error Index

[m | a-d][1-8]662

The checksum for the contents of the EEPROM in head 1 did not match the checksum stored in the EEPROM.

Error Index ????

The first EEPROM word of each head was $FFFF_{hex}$. Both heads are therefore assumed to be "not present".

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()

Routine Name Calibration

Overview This exerciser performs calibration of the acquisition system and reports calculated calibration constants and any errors encountered. This exerciser allows running a selected part of calibration on one acquisition channel or running full calibration on both acquisition channels.

Description The exerciser displays a menu from which one of five calibration functions can be selected:

1. Comparator reference
2. Signal offset
3. Offset correction
4. Gain
5. Loop on all parts for both channels until error occurs

Functions 1 through 4 perform a selected part of calibration on one acquisition channel. When one of these choices is made, the user is prompted to enter the channel to be calibrated. Function 5 does a full calibration on both acquisition channels, looping until a calibration error occurs or the operator aborts the test.

For each of the above functions, the calibration is performed and the resulting calibration constants are displayed. When errors occur, and only when errors occur, error messages are displayed (as indicated by messages marked with ** below). The calibration constants and error messages are described as follows for each calibration function:

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]
Acq[1-8]
Acq[1-8]
Acq[1-8]
Acq[1-8]

Exercisers
Exercisers
Exercisers
Exercisers
Exercisers

Calibration ()
Calibration ()
Calibration ()
Calibration ()
Calibration ()

COMPARATOR REFERENCE

Message: $\text{intcpt}[\text{POS}] = w$ $\text{slope}[\text{POS}] = x$
 $\text{intcpt}[\text{NEG}] = y$ $\text{slope}[\text{NEG}] = z$

Description: Slope and intercept for linear functions converting 0.25mv steps into comparator reference DAC codes for both positive (POS) and negative (NEG) transitions.

w = intercept for positive transition function
(uncalibrated default = 8192)
x = slope for positive transition function
(uncalibrated default = 0.97524)
y = intercept for negative transition function
(uncalibrated default = 8192)
z = slope for negative transition function
(uncalibrated default = 0.97524)

Message: ****err_id = 0 start = x level = y**

Description: Could not get comparator output to correct state to start search for cross-over point.

x = initial DAC code for comparator reference DAC
y = DAC code limit that was hit

Message: ****err_id = 1 start = x level = y**

Description: Could not get comparator output to switch state in binary search phase.

x = initial DAC code for comparator reference DAC
y = DAC code limit that was hit

Message: ****err_id = 2 start = x level = y**

Description: Could not get comparator output to switch state in linear search phase.

x = initial DAC code for comparator reference DAC
y = DAC code limit that was hit

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()

SIGNAL OFFSET

Message: x2_intcpt = w x2_slope = x
 x20_intcpt = y x20_slope = z

Description: Slope and intercept for linear functions converting
 0.25mv steps into signal offset DAC codes for both x2
 gain setting and x20 gain setting.

w = intercept for x2 gain function
 (uncalibrated default = 8192)
 x = slope for x2 gain function
 (uncalibrated default = 0.97524)
 y = intercept for x20 gain function
 (uncalibrated default = 8192)
 z = slope for x20 gain function
 (uncalibrated default = 0.97524)

Message: **err_id = 8 level = x out_low = y out_hi = z

Description: A/D reading for low or high gain setting was under-
 range or over-range.

x = signal offset DAC code
 y = A/D reading for low gain setting
 z = A/D reading for high gain setting

Message: **err_id = 9 level = x out_low = y out_hi = z

Description: Signal offset DAC was set over max code or under min
 code.

x = signal offset DAC code
 y = A/D reading for low gain setting
 z = A/D reading for high gain setting

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()

OFFSET CORRECTION

Message: x2_oscor_zero = x1 x2 x3 x4 x5 x6
 x20_oscor_zero = y1 y2 y3 y4 y5 y6

Description: Offset correction DAC settings for 6 different variable gain DAC settings for both x2 gain setting and x20 gain setting. The function that sets the offset correction DAC is a piece-wise linear function that is dependent on the variable gain DAC setting.

x1,..., x6 = offset correction DAC codes for x2 gain setting (uncalibrated defaults = 128)

y1,..., y6 = offset correction DAC codes for x20 gain setting (uncalibrated defaults = 128)

Message: **err_id = 16 level = x ad_out = y cnt = z

Description: Offset correction DAC was set over max code or under min code.

x = offset correction DAC code

y = A/D reading

z = number of A/D readings within range of 127 to 129

Message: **err_id = 17 level = x ad_out = y cnt = z

Description: A/D reading was under-range or over-range.

x = offset correction DAC code

y = A/D reading

z = number of A/D readings within range of 127 to 129

GAIN

Message: x2_net_gain = x1 x2 x3 x4
 x20_net_gain = y1 y2 y3 y4

Description: Total signal path gain for 4 different variable gain DAC settings for both x2 gain setting and x20 gain setting. The function that sets the variable gain DAC is

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()
Acq[1-8]	Exercisers	Calibration ()

a piece-wise linear function that is dependent on the total signal path gain. Total signal path gain is a constant function of vertical sensitivity in mv/div.

x1,..., x4 = total gains for x2 gain setting
(uncalibrated defaults = 1.0, 2.0, 4.0, 10.0)
y1,..., y4 = total gains for x20 gain setting
(uncalibrated defaults = 10.0, 20.0, 40.0, 100.0)

Message: **err_id = 24 level = x out_low = y out_hi = z

Description: A/D reading for low or high input was under-range or over-range.

x = signal offset, in mv (used as input source)
y = A/D reading for -x as input
z = A/D reading for +x as input

Message: **err_id = 25 level = x out_low = y out_hi = z

Description: Same as above, but input source is combination of +16 mv calibration input and signal offset.

x = signal offset, in mv (used as part of input source)
y = A/D reading for -x as input
z = A/D reading for +x as input

Error Index None.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()

Routine Name DACs

Overview This exerciser allows setting, ramping, or stepping any of the analog control DACs on the analog board.

Description The exerciser displays a menu of 9 DACs that can be exercised:

1. Comparator Reference
 (Comparator Control Voltage U711 and U710 <10>)
2. Signal Offset
 (Offset Control Voltage U811 and U810 <10>)
3. Offset Correction
 (Offset Correction & TDR Delay Control Voltage U1200 <10>)
4. Variable Gain
 (Prog Gain Amp Gain Control Voltage U1201 <10>)
5. S/H Function 1
 (Sampling Head Blowby Control Voltage U1101 <10>)
6. S/H Function 2
 (Sampling Head Loop Gain Control Voltage U1100 <10>)
7. S/H Function 3
 (Sampling Head Offset Control Voltage U901 and U900 <10>)
8. S/H Function 4
 (Sampling Head TDR Level Control Voltage U1000 <10>)
9. S/H Function 5
 (Offset Correction & TDR Delay Control Voltage U1200 <10>)

Once the DAC to exercise is chosen, the available exercise functions are displayed:

1. Set to fixed value -
 Allows the user to enter a decimal DAC code at which the DAC will be set. For the **Comparator Control Voltage U711 and U710 <10>** and **Offset Control Voltage U811 and U810 <10>** DACs,

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()
Acq[1-8]	Exercisers	DACs ()

the user can enter either the DAC code or set the DAC to a specific setting in mv (using calibration constants).

2. Ramp -
The DAC is ramped from 0 to its maximum value, and then back to 0 again. This sequence repeats until the user exits the exerciser.
3. Step -
The user is prompted for the two DAC codes to step the DAC between. The DAC is stepped between the two codes until the user exits the exerciser.

After the exercise function is chosen, the user is prompted for the specific channel to exercise. For the first 4 DACs, the channel will be either "a" or "b" (for one of the two acquisition channels), and for the sampling head DACs, the channel will be "1", "2", "3", or "4" (for one of the four sampling head channels).

Error Index

None.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()

Routine Name Signal Path

Overview This exerciser allows the user to configure the analog signal path from the input (**Chan Input Mux A** or **Chan Input Mux B** <9>) to the A/D (**Chan A/D A** or **Chan A/D B** <9>). Functions supported include connecting an input to the channel, setting gain and offset, and generating strobes and looking at the output of the A/D.

Description The exerciser displays a menu from which one of seven choices can be made:

1. Connect input to acq. channel
2. Set offset
3. Set gain
4. Generate strobes
5. Constant strobe & display A/D output
6. Display waveform memory
7. Fill waveform memory

The following describes the seven available functions:

CONNECT INPUT TO ACQ. CHANNEL

The user is prompted to connect one of the eight inputs to one of the acquisition channels. The eight available inputs are:

1. External channel 1
2. External channel 2
3. External channel 3
4. External channel 4
5. +2 volt cal.
6. -2 volt cal.
7. +16 mv cal.
8. Ground

The **Chan Input Mux A** or **B** U301 or U100 <9> is set to the selected input by setting bits 15, 14, 13 in the following manner:

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()

	15	14	13
External channel 1	H	H	H
External channel 2	H	H	L
External channel 3	H	L	H
External channel 4	H	L	L
+2 volt cal.	L	H	H
-2 volt cal.	L	H	L
+16 mv cal.	L	L	H
Ground	L	L	L

SET OFFSET

The user is prompted to set an offset voltage in millivolts for the **Offset Control Voltage** DAC U811 or U810 <10> for one of the acquisition channels. The offset voltage is converted to DAC codes using calibration constants and then the appropriate channel's DAC is updated.

SET GAIN

The user is prompted to set the gain in mv/div for one of the acquisition channels. The x2/x20 gain switch (Chan Mux Latch U1501 pin 13 or 12 <9>) and **Prog Gain Amp Gain Control Voltage** DAC U1201 <10> for the selected channel is updated using calibration constants. For gains from 255 to 20 mv/div, the x2/x20 switch is set for x2 mode (high) and for gains from 19 to 2 mv/div, the switch is set for x20 mode (low). The variable gain voltage goes from about 250 mv (at 200 mv/div and 19 mv/div) to about 2.5 volts (at 20 mv/div and 2 mv/div).

GENERATE STROBES

This function generates local strobes and acquires data. The user is first prompted to select the area into which the acquired data will be put. The data can be acquired into either normal waveform or calibration waveform areas. The user is also prompted for the number of strobes to generate and whether to do continuous or one-shot strobing.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()

- Enable Chan A/D A and B <9> by setting **Measurement & A/D Strobe Generator U901** pins 16 and 17 <13> high.
- Enable the processor-generated strobe (MPUSTB) by setting **Enable Register U801-18 MSTREN(L)** <13> low.
- Clear the waveform address counter by pulsing **Hardware Measurement System U820-13 WACCLR(L)** <13>.
- If a sampling head simulator is installed in channel 1/2 of the acquisition system, clear its waveform address counter by setting **Sampling Head Digital Controls U1500-17** high and then low.
- If a sampling head simulator is installed in channel 3/4 of the acquisition system, clear its waveform address counter by setting **Sampling Head Digital Controls U1500-15** high and then low.
- If the user elected to collect waveform data into the calibration waveform area, pulse **Hardware Measurement System U820-12 WACSET(L)** <13>.
- Generate the number of strobes selected by the user by pulsing **Interrupt Latch U1100B-5 MPUSTB** <13>.
- If the user selected one-shot strobing, exit the exerciser.
- For continuous strobing, the WAC clear and strobing sequence is repeated until the user exits the exerciser.
- When the exerciser is exited, the processor-generated strobes are disabled by setting **Enable Register U801-18 MSTREN(L)** <13> high and **Chan A/D A <9>** and **Chan A/D B <9>** are disabled by setting **Measurement & A/D Strobe Generator U901** pins 16 and 17 <13> low.

CONSTANT STROBE AND DISPLAY A/D OUTPUT

This function constantly resets the waveform address counter, generates a local strobe, and displays the A/D reading. The user is prompted for the acquisition channel to exercise.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()

- Enable the processor-generated strobe (MPUSTB) by setting **Enable Register U801-18 MSTREN(L) <13>** low.
- If the user selected acquisition channel A, enable **Chan A/D A <9>** by setting **Measurement & A/D Strobe Generator U901-16 <13>** high.
- If the user selected acquisition channel B, enable **Chan A/D B <9>** by setting **Measurement & A/D Strobe Generator U901-17 <13>** high.
- If a sampling head simulator is installed in channel 1/2 of the acquisition system, clear its waveform address counter by setting **Sampling Head Digital Controls U1500-17** high and then low.
- If a sampling head simulator is installed in channel 3/4 of the acquisition system, clear its waveform address counter by setting **Sampling Head Digital Controls U1500-15** high and then low.
- The following steps are repeated until the user exits the exerciser. (Note that the sampling head simulator is not sent a waveform address clear in this loop in order to allow cycling through the simulated waveform data.)
 - Clear the waveform address counter by pulsing **Hardware Measurement System U820-13 WACCLR(L) <13>**.
 - Generate a strobe by pulsing **Interrupt Latch U1100B-5 MPUSTB <13>**.
 - Read location *0003hex* (for channel A) or *0002hex* (for channel B) and display the value on the screen.
- When the exerciser is terminated, the processor-generated strobes are disabled by setting **Enable Register U801-18 MSTREN(L) <13>** high and **Chan A/D A <9>** and **Chan A/D B <9>** are disabled by setting **Measurement & A/D Strobe Generator U901** pins 16 and 17 <13> low.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()
Acq[1-8]	Exercisers	Signal Path ()

DISPLAY WAVEFORM MEMORY

This function displays either normal waveform or calibration waveform data.

FILL WAVEFORM MEMORY

This function fills either normal waveform or calibration waveform data area with a fill byte that the user enters.

Error Index

None.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()

Routine Name Meas Sys (Measurement System)

Overview This exerciser allows checking various parts of the measurement system through two functions that the user can select. One function generates a test waveform that allows both channels' comparators, dot filters, and transition detectors to be checked. The other function ramps the comparator reference voltage and strobes the comparator to see where the comparator output switches state. It then calculates what the input voltage is, based on calibration constants.

Description The exerciser displays a menu from which one of two choices can be made:

1. Generate meas. system test waveform
2. Calculate input voltage

The following describes the two available functions:

GENERATE MEAS. SYSTEM TEST WAVEFORM

This function generates a test waveform that allows checking both channels' comparators, dot filters, and transition detectors. The comparator outputs are alternately set high and low for increasing strobe counts.

Perform set-up:

- Both channels' comparators and dot filters are enabled and A/Ds are disabled by setting **Measurement & A/D Strobe Generator** U901 <13> pins 19, 18, 15 and 14 high and pins 17 and 16 low.
- Enable the processor-generated strobe (MPUSTB) and disable transition found interrupts by setting **Enable Register** U801-18 MSTREN(L) <13> low and pins 16 and 15 (TAINTEN and TBINTEN) low.
- Set **Comparator Control Voltage** U711 and U710 <10> outputs (VCREFA and VCREFB) to ground.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Meas Sys 0
Acq[1-8]	Exercisers	Meas Sys 0
Acq[1-8]	Exercisers	Meas Sys 0
Acq[1-8]	Exercisers	Meas Sys 0
Acq[1-8]	Exercisers	Meas Sys 0

Loop on the following steps until the user exits the exerciser:

- Set Chan Mux Latch U1501 <9> pins 14, 16, 17 and 19 low and pins 15 and 18 high to select -2V REF at Chan Input Mux A U301-3 <9> and at Chan Input Mux B U100-3 <9>.
- Pulse A Meas Comparator U400-6 CMPSTBA <9> and B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) 128 times.
- Pulse Hardware Measurement System U820-2 DLACLR and U820-63 DLBCLR <13> to clear Dot Filter Counter A and Dot Filter Counter B.
- For an inner loop 24 times, perform the following items:
 - Set Chan Mux Latch U1501 <9> pins 14 and 17 low and pins 15, 16, 18 and 19 high to select +2V REF at Chan Input Mux A U301-4 <9> and at Chan Input Mux B U100-4 <9>.
 - Pulse A Meas Comparator U400-6 CMPSTBA <9> and B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) "inner loop count" times.
 - Set Chan Mux Latch U1501 <9> pins 14, 16, 17 and 19 low and pins 15 and 18 high to select -2V REF at Chan Input Mux A U301-3 <9> and at Chan Input Mux B U100-3 <9>.
 - Pulse A Meas Comparator U400-6 CMPSTBA <9> and B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) "inner loop count" times.

When the user exits the exerciser, the processor-generated strobes are disabled by setting Enable Register U801-18 MSTREN(L) <13> high

CALCULATE INPUT VOLTAGE

This function ramps the comparator reference voltage for one acquisition channel both up and down and strobes the comparator to see where the comparator output switches state for the two ramp

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()

directions. It then calculates and displays what the input voltage is based on calibration constants.

Perform set-up:

- If channel A was selected, enable that channel's comparator and disable channel B comparator by setting **Measurement & A/D Strobe Generator U901 <13>** pin 18 high and pin 19 low.
- If channel B was selected, enable that channel's comparator and disable channel A comparator by setting **Measurement & A/D Strobe Generator U901 <13>** pin 19 high and pin 18 low.
- Disable both channels' A/D by setting **Measurement & A/D Strobe Generator U901 <13>** pins 16 and 17 low.
- Enable the processor-generated strobe (MPUSTB) by setting **Enable Register U801-18 MSTREN(L) <13>** low.

Loop on the following steps until the user exits the exerciser:

- For first inner loop (max of 16383 times) with loop count going up, perform the following items:
 - Set **Comparator Control Voltage <10>** for selected channel to "inner loop count" codes.
 - Pulse **A Meas Comparator U400-6 CMPSTBA <9>** or **B Meas Comparator U200-6 CMPSTBB <9>** (via Interrupt Latch U1100B-5 MPUSTB <13>) once.
 - If **A/D Status Register U800-2 COMPA <13>** (for channel A) or **A/D Status Register U800-6 COMPB <13>** (for channel B) is low, break inner loop since comparator output has switched state. Store reference DAC code at which this switch occurred.
- For second inner loop (max of 16383 times) with loop count going down, perform the following items:
 - Set **Comparator Control Voltage <10>** for selected channel to "inner loop count" codes.

Main Acq
 MCU A Acq
 MCU B Acq
 MCU C Acq
 MCU D Acq

Acq[1-2]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()
Acq[1-8]	Exercisers	Meas Sys ()

- Pulse A Meas Comparator U400-6 CMPSTBA <9> or B Meas Comparator U200-6 CMPSTBB <9> (via Interrupt Latch U1100B-5 MPUSTB <13>) once.
- If A/D Status Register U800-2 COMPA <13> (for channel A) or A/D Status Register U800-6 COMPB <13> (for channel B) is high, break inner loop since comparator output has switched state. Store reference DAC code at which this switch occurred.
- Calculate what input voltage both cases of DAC codes represents and display on screen.

When the user exits the exerciser, the processor-generated strobes are disabled by setting Enable Register U801-18 MSTREN(L) <13> high

Error Index None.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

Acq[1-2]	Exercisers	Registers ()
Acq[1-8]	Exercisers	Registers ()
Acq[1-8]	Exercisers	Registers ()
Acq[1-8]	Exercisers	Registers ()
Acq[1-8]	Exercisers	Registers ()

Routine Name Registers

Overview This exerciser allows setting up the path to, and the reading of, the diagnostics A/D and displaying the contents of the sampling head EEPROMs.

Description The exerciser displays a menu from which one of two choices can be made:

1. Set diag. A/D input and read diag. A/D
2. Display sampling head EEPROM contents

The following describes the two available functions:

SET DIAG. A/D INPUT AND READ DIAG. A/D

This function displays and allows loading of the diagnostics A/D input select register and then starts reading and displaying the diagnostics A/D.

- Read and display contents of **Diagnostic Mux Control Latch U1401 <11>**.
- Prompt user for new value for **Diagnostic Mux Control Latch U1401 <11>** and write it to the latch. Read and display latch again as a verification step.
- Loop on the following items until the user exits the exerciser:
 - Read diagnostics A/D by writing to address **7000_{hex}**. This causes **Diagnostic A/D U1410-15 <11>** to go low to start a read of the A/D. The status of the A/D conversion is monitored by reading **A/D Status Register U1300-18 <11>**. When this bit goes high, the conversion is done and address **7000_{hex}** is read. This causes U1400 pins 1 and 19 <11> to go low, which puts the A/D results on the data bus.
 - Display 16 A/D readings to a line on the screen.

Main Acq
MCU A Acq
MCU B Acq
MCU C Acq
MCU D Acq

